GDAŃSK UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRONICS, TELECOMMUNICATIONS AND INFORMATICS

The author of the PhD dissertation: **Adam Blokus**
Scientific discipline: Computer Science

DOCTORAL DISSERTATION

Title of PhD dissertation: *Image classification based on video segments*

Title of PhD dissertation (in Polish): *Klasyfikacja obrazów na podstawie fragmentów strumienia wideo*

Supervisor *signature*
prof. dr hab. inż. Henryk Krawczyk, prof. zw. PG

Gdańsk, year 2018

Gdańsk, 12.03.2018

Image classification based on video segments

mgr inż. Adam Blokus

Abstract

In the dissertation a new method for improving the quality of classifications of images in video streams has been proposed and analyzed. In multiple fields concerning such a classification, the proposed algorithms focus on the analysis of single frames. This class of algorithms has been named OFA (One Frame Analyzed).

In the dissertation, small segments of the video are considered and each image is analyzed in the context of its closest neighborhood, which is defined by a shifting time window. The class of algorithms representing such an approach has been named FSA (Frame Sequence Analyzed).

Experiments on a number of video streams of different types have confirmed that the FSA method improves the classification results by reducing the level of error on average by 20%.

Two variants of FSA algorithms have been analyzed: iFSA – which considers only OFA decisions, and fFSA – which considers OFA decisions as well as the similarity between the analyzed frames. Furthermore, the variants differ in terms of their computational complexity.

The analysis of the proposed FSA algorithms included different configurations of decision functions, multiple similarity measures, as well as method parameters such as: the window size, the significance weight distribution parameter or the decision acceptance threshold. The FSA algorithms have been evaluated in terms of those attributes, which has proven their applicability in terms of the type and intensity of distortions in the video stream. Furthermore, the performed tests have confirmed the effectiveness and versatility of the FSA method.

Gdańsk, 12.03.2018

Klasyfikacja obrazów na podstawie fragmentów strumienia wideo

mgr inż. Adam Blokus

Streszczenie

W rozprawie zaproponowano i przeanalizowano nową metodę zwiększającą jakość klasyfikacji obrazów w strumieniach wideo. W wielu dziedzinach dotyczących takiej klasyfikacji, proponowane algorytmy koncentrują się na analizie pojedynczych klatek. Tego rodzaju klasę algorytmów nazwano OFA (ang. One Frame Analyzed).

W rozprawie doktorskiej rozpatrzono sekwencje obrazów, analizując każdy obraz w kontekście jego najbliższego otoczenia, określanego za pomocą przesuwnego okna czasowego. Klasę algorytmów związaną z takim podejściem nazwano FSA (ang. Frame Sequence Analyzed).

Jak wykazały eksperymenty dotyczące analizy różnego typu strumieni wideo, metoda FSA zwiększa jakość klasyfikacji, poprzez ograniczenie poziomu błędów średnio o co najmniej 20%.

Rozpatrzono dwa warianty algorytmów FSA: iFSA – uwzględniający tylko decyzje OFA oraz fFSA – uwzględniający decyzje OFA oraz dodatkowo podobieństwa występujące między analizowanymi klatkami. Poza tym te klasy algorytmów różnią się złożonością obliczeniową.

W proponowanych algorytmach FSA uwzględniono różne konfiguracje funkcji decyzyjnych, różne miary podobieństwa, jak również takie parametry jak: szerokość okna, parametr rozkładu wag ważności decyzji OFA w oknie, czy próg akceptowalności). Przebadano jakość algorytmów FSA w zależności od tych konfiguracji, wykazując adekwatność ich zastosowania w zależności od typu i poziomu zakłóceń strumienia wideo. Poza tym, przeprowadzone testy potwierdziły skuteczność i uniwersalność metody FSA.

*I would like to thank: my supervisor for
our long cooperation and numerous discussions,
my family for the constant encouragement,
and Ola for her tremendous support and patience.*

Contents

1 Introduction

Multimedia systems have been a very wide area of research for a long time – and they are finding their way into a constantly growing list of domains due to the steadily increasing computing power of modern devices and emergence of new applications. A vital component of multimedia are video streams, whose processing and understanding brings a lot of challenges and is a source of numerous research subjects. Multiple applications can be listed for which new ways of gathering, processing and interpreting video data are being developed. They range from security (CCTV monitoring, facial recognition), telecommunication, through autonomous vehicles analyzing their surroundings, quickly digitalized scans of designs and plans, to applications in various medical fields which recognize and classify lesions and other visible features.

Regarding the field of classifying images in video streams, two independent approaches can be taken for further improving the quality of classifications: applying constantly emerging methods classifying single images or treating the video stream as a whole and classifying the frames in their temporal context. A representative of the first category are the currently very popular deep feed-forward neural networks (e.g. convolutional). The second category contains approaches such as recurrent layers in neural networks or hidden Markov models (HMMs). Every approach has to be analyzed in terms of its universality, training costs and provided efficiency.

While multiple propositions and approaches emerge all the time for further improving and innovating the classification, they often apply complex tools and theories for application in narrow fields. One of the motivations of this work is to develop an approach which would allow to improve classification results in a wide range of domains. The analysis of existing literature, related to this subject by various aspects, indicates that such a research question has not been addressed thus far. Besides single works, where authors relied only on simple intuitions, this dissertation is among the first, which tackle the problem of improving existing classifications by incorporating the temporal context of the video.

In this work an original universal method is proposed for binary classifications in video streams, which builds on top of all of the aforementioned approaches. It bases on the idea that the output of any algorithm classifying single frames can further be improved when perceived as a temporal sequence representing a continuous process. We will further call it FSA (Frame Sequence Analysis). It allows to improve the quality of classification and requires only a limited set of assumptions to be fulfilled, regarding the continuity of the video stream and sufficient performance of the underlying algorithm.

This work has been supported by *The Centre Of Competence For Novel Infrastructure Of Workable Applications* European project POIG.02.03.00-22-059/13-00

Furthermore, the method does not operate on the whole video, but on short **video segments** – contiguous sequences of frames within the video, processed in a shifting-window approach. This allows the classification to be easily distributed, contrary to methods like recurrent neural networks or HMMs.

The FSA method is based on a two step scheme - classifying frames in a video by an underlying algorithm (called OFA), followed by a post-processing step (the FSA step), which intends to rationalize the result, taking into consideration that it represents a continuous process. This is performed by considering a shifting window, centered on consecutive frames. The contents of the window provide the temporal context required for the reasoning in the FSA step. At the same time the processing of a single frame remains isolated from the whole video, which is the key feature for the aforementioned easy parallelization.

The proposed method comes in two variants: iFSA and fFSA. It will be shown that both introduce a significant improvement of the quality of classification of frames in the video. They are also very efficient in terms of computation time - especially in the case of iFSA the overhead is imperceptible.

Although only binary classifications are handled by the proposed method, possible extensions for multi-categorical classifications can be proposed in a manner analogous to those applied e.g. in neural networks.

Another important feature of the proposed approach is its low cost of training and application. Even though an optimal parameterization can be established for every domain and improved algorithm, there are ranges of parameters which provide significant improvement out-of-the-box. The tuning of any FSA method is independent of the training of the underlying algorithm, which significantly reduces the cost of preparing an appropriate training set of whole video segments: The training can be performed on an image dataset, requiring only a small number of videos for establishing the best FSA parameters.

The dissertation is organized as follows: In Chapter 2 a wide overview of related works has been presented. Due to the originality of this dissertation, most of the cited works only implicitly apply or discuss methods which motivated the FSA approach. The chapter ends with a general vision of the proposed solution and thesis statements.

Chapter 3 describes the analytical basis of this work. Terms such as video classification and continuity are formally defined and conclusions are drawn regarding the applicability of the proposed approach in real-life videos.

Chapter 4 builds on the previous chapter and proposes for the FSA method a concrete algorithm and its building blocks, together with additional extensions. This chapter ends with a definition of the proposed evaluation criteria of the classification results. Besides simple accuracy measures, also measures for evaluating the scene segmentation stability are proposed.

Next, Chapter 5 presents all elements of the testing procedure for the proposed method. First, the test data and testing environment are described. Afterwards, value ranges for parameters of the FSA algorithm are established by analyzing the properties of test data. Finally, three

decision functions are proposed for further evaluation.

The results of all experiments have been summarized in Chapter 6. Consecutive sections contain experiments addressing the general validity of the FSA approach and all parts of the thesis statements.

Finally, the acquired results are concluded in Chapter 7, where also directions for further research are described.

2 Problem description

In this chapter we present the motivations for the presented research and the fields in which it might find use. After discussing a number of diverse works implicitly and explicitly related to the subject of image classification in video segments, we arrive at the conclusion that a common general approach can be defined, which results in a new, original and universal method. In the end, a number of statements about the expected performance of the newly defined Frame Sequence Analysis (FSA) approach are presented.

2.1 Motivation

With the radical development of optical devices and methods for automatic image classification new approaches of applying them in real life usages start to emerge. Such trends are especially visible in medicine (diagnostics and monitoring) or entertainment (e.g. facial recognition in consoles with Kinect cameras [91, 172]) but can also be found in fields such as automated video surveillance, meteorology or traffic monitoring. We consider algorithms which are processing video frames to detect problem-specific features. Depending on the particular nature of each application, the sought features are either **static** (recognizable on a single picture, e.g. the color of blood in a medical picture, number of faces in a CCTV frame) or **dynamic** (e.g. the tracked object's movement in video surveillance, measured heart rate).

As presented in Figure 2.1, the dynamic features are those which have a time component to them, are related to variability in time, or require to persist a state (e.g. a counter of entries to a room) over the course of processing. For instance, describing various parameters of movement - direction, pace, acceleration - involves specifying the change of position or change of pace in a unit of time. Static features are observable in single snapshots of the recorded view, where no variety is described (but not necessarily not present). For example, in the case of moving objects, a point in time has to be specified to ask for an objects position. Once the recording is frozen in that point, it is still possible to determine the position of an object, but not the pace of it anymore (which would require acquiring at least one more position information in a different point in time) or acceleration (which requires three different points altogether).

While detecting a static feature requires only a single picture (frame), in real-life videos we might also consider a longer sequence of frames where the same feature should be visible. For example, lesions in endoscopic examination videos can be recognized as a static element in a single picture but their visibility can be expected to last for a number of consecutive frames. Similarly, a license plate seen on a surveillance recording remains visible as long as the vehicle

(a) A single frame/image allows to detect static features. Those can be for instance:
- the presence of a person,
- the position of the person in their surroundings,
- the person's identity,
- lighting conditions in the environment.

(b) A sequence of frames allows to detect, besides static, also dynamic features. Those can be for instance:
- if a person is moving or not,
- the speed and direction of moving objects,
- the number and identities of people who entered a room in a given period.

Figure 2.1 – Examples of static (a) and dynamic (b) features in a real-life video.

remains in sight and in the right position relative to the camera. It might happen that at some frames the license number is incomprehensible - but all changes in visibility happen over time, at a pace influenced by the movement of the observed objects.

A camera is recording consecutive frames of the video at a given rate. The frames are pictures of the camera's **view** at the corresponding points of time. Changes of the video can be both the result of events and object transitions within the view, as well as a shift of the whole view related to the movement of the camera.

Knowing the characteristics of a particular video source (especially its pace of change, e.g. the velocity of a camera moving in the filmed environment), it should be possible to establish a frame rate at which consecutive frames remain similar, i.e. in most cases their differences do not exceed an arbitrarily given threshold. A video stream in which such an assumption is true will further be called **continuous**.

In a continuous stream we can expect that a static feature's presence will extend over a sequence of multiple frames. Such a sequence of consecutive frames sharing the static property

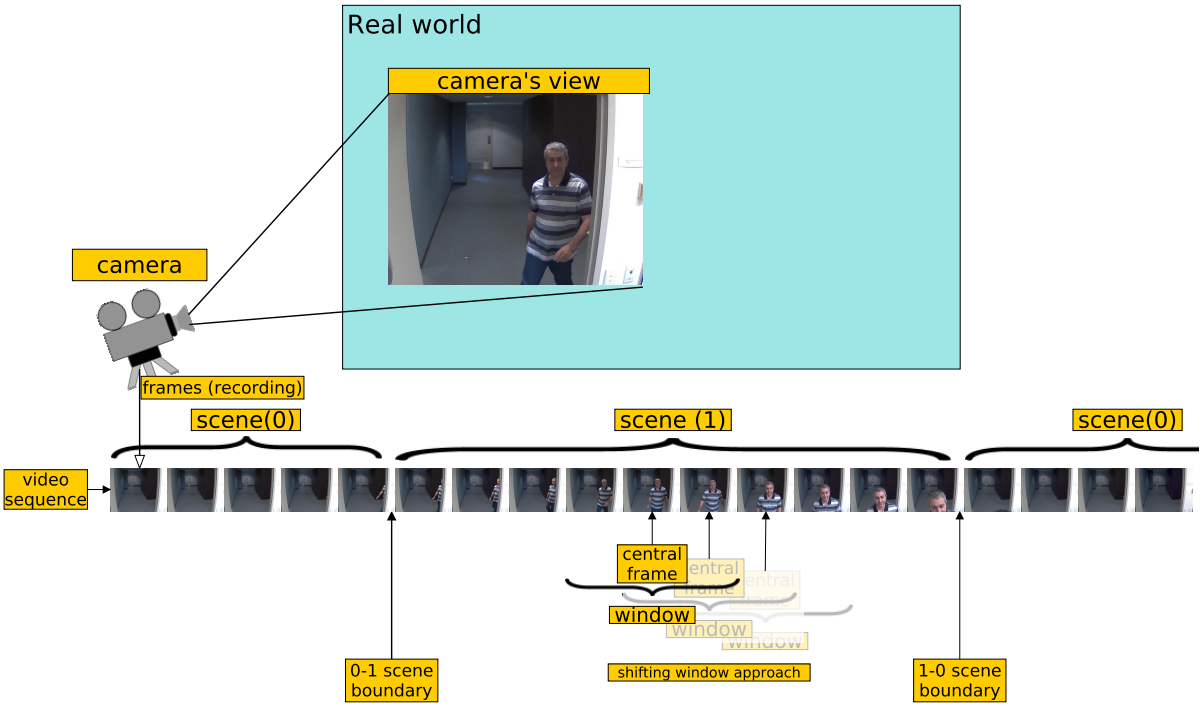


Figure 2.2 – Illustration of elementary concepts.

of our interest is called a **scene**[1]. Some ambiguity is expected and acceptable during scene transitions, e.g. an object can be only partially visible when entering the view.

The concepts described so far in this section have been illustrated in Figure 2.2. A portion of the real world falls within the view of the camera. The video sequence contains the consecutive pictures (frames). In the case of binary classifications, the scenes are alternately negative (0) and positive (1).

The approach proposed in this work uses small shifting time-windows, which are much smaller than the considered scenes. This ensures that the vast majority of considered windows is contained fully within single scenes, and only a limited number of windows overlaps a scene boundary. This in turn leads to the assumption that the shared class of frames within the windows can be used to increase the reliability of static feature detection – and also the quality of the resulting segmentation into scenes.

A wide set of algorithms susceptible to such an improvement are those developed for Wireless Capsule Endoscopy (WCE, [71]). In this diagnostic method cameras contained in swallowable capsules collect hours of video material that a medical doctor has to look through and classify. This task can take even up to 2 hours of a specialist's time [147] unless they are supported by additional tools. One of them are algorithms recognizing lesions and bleedings [26] in the video stream. The doctor gets a summary of the potentially interesting scenes from the video, what

[1]The term "scene" has two meanings in video processing: a subdivision of a movie/play or a stage setting. Within this work, the first meaning is used. For clarity, the latter has been named **view**.

allows them to focus on these regions first. Furthermore, if a proper processing performance is ensured, such algorithms can be applied in real-time in the course of classic examinations in order to support the medical doctor in their decision making [16].

A similar distinction of applications can be found in the processing of videos recorded by CCTV, personal cameras or driver support systems in cars. Long films can be processed for detecting the scenes with the presence of particular people of interest ([93, 127, 169]). Other cases can be conceived, e.g. when an audit of all activities in a monitored area is performed on the basis of stored recordings. If scenes with a presence of people could be identified automatically, the auditor's duties could be limited to a review of those findings. In the case of traffic recordings, particular events or conditions might be sought in archival data to validate the proper behavior of autonomous cars in given conditions. And again, if sufficient computing power and a low latency algorithm are provided, the videos can be processed in real-time to alert about ongoing events.

The two different kinds of algorithm applications lead us to distinguishing two corresponding processing models of algorithms which classify video frames: offline and online (with a possible delay). A basic outline of those approaches has been presented in Table 2.1.

Within this dissertation we focus primarily on the **offline** processing model. Videos processed offline are acquired not directly from the camera, but from any kind of data storage. All of the frames are available instantly and the possible amount of video available for processing is practically unlimited. Therefore processing on distributed systems has to be considered, since a single node might not be able to handle the stored data in reasonable time. The evaluation of the computational efficiency of offline algorithms is focused on their throughput, i.e. amount of data they are able to process in a unit of time.

Online algorithms process consecutive incoming frames and provide classifications in real-time or with a delay. The delay can be caused by the algorithm itself or be imposed by a specific hardware architecture. Videos processed online are acquired frame-by-frame directly from a camera or in an equivalent manner (e.g. online stream or tape media). For classifying any given frame only its preceding sequence is available. A possible inclusion of succeeding frames results in a corresponding delay. Real-time applications require the use of online processing algorithms. The properties defining the applicability of a given algorithm in terms of computational efficiency are its throughput and latency. It is worth noting that the offline method proposed in this dissertation is also suitable for applying it as an online algorithm with a fixed delay.

After defining the processing models, we will finish the section with an introduction of the OFA algorithms and the ground for their improvement.

Methods which classify single images can be iteratively applied to consecutive images of the video. The acquired sequence of classification results can be afterwards presented as the final classification result of the whole video. Within this dissertation algorithms operating in this manner have been named OFA (One Frame Analyzed) algorithms.

Table 2.1 – Processing models for image classification in video streams and improvement of OFA classifications.

Processing model	**Outline**	**Considered frames**
Online	For every consecutive frame index m : Input: frame/OFA classification m Output: final classification for frame m	All until current frame
Online with delay	For every consecutive frame index m: Input: frame/OFA classification m Output: final classification for frame $m-delay$	All until current frame and frames within the period of *delay* frames after it.
Offline	Input: all frames/OFA classifications Output: all final classifications	All, or all within part of video if video processing is distributed.

Such an approach has the advantage of not requiring any development of methods dedicated for videos and allowing to only incorporate image classification methods. The field of image classification is extensively researched and numerous methods are available on the spot. Both the algorithms and the available training sets can remain unchanged and only two additional layers for splitting the video into single images and combining the acquired results into a sequence are required. The OFA approach has a number of significant advantages:

1. Simple parallelization opportunities are evident as the processing of different frames can be done independently on separate computing nodes.
2. If the pace of change is too high to consider the video continuous, it cannot be treated as anything more than just loosely correlated pictures anyway.
3. Building datasets of single pictures is much cheaper than labeling complete videos frame by frame.

Nevertheless, classifying single video frames without considering the whole video as a sequence would be fully justified only if the incoming video sequence could be treated as a set of unrelated pictures. At the current level of technical development, in most domains cameras create videos where neighboring frames are highly related with each other – what can be observed even for such typical recording rates as 25 FPS. Even in the field of WCE, which at first used cameras recording at 2-3 FPS [157], there are models able to record at 30 FPS [109, 123].

In principle, the independent classification of frames in the OFA approach could lead to acquiring proper results (i.e. if a perfect classification algorithm was to be used). Still, due to the inevitable imperfection of every artificial intelligence-related method of image recognition, various additional problems emerge. Occasional mistakes result in segmenting sequences of frames of one kind (i.e. presenting lesions or healthy tissue), and therefore impede some possible ways of presenting the results and a proper summarization of the classification process.

Furthermore, minor glitches in the video stream (such as artifacts, flashes or magnetic disturbance) are prone to misclassification if the information from the temporal neighborhood of such effects is not being considered.

If a relatively high accuracy level of the basic algorithms (i.e. those which classify single frames) is maintained, the aforementioned issues can be addressed by the construction of new methods for post-processing the sequence of classifications. In such an approach, besides the single results, also the temporal properties of the frame sequence can be considered as important information. We call the process of changing preliminary results so that they correspond more closely to the expected model of the observed process **result rationalization**.

In the next sections we will discuss research done in this field as well as other results acquired mostly with the intuitive application of similar kinds of reasoning. Some focus will also be put on the bibliography related to methods which establish frame similarity, which can be closely related to the temporal parameters of the video stream.

2.2 Related Publications

Altogether, the topic of rationalizing classification results has not been directly and fully addressed so far by other authors. The considerations and experiments in this dissertation have been influenced and overlap with the works of Blokus et al., of which the most important are works which initiated the topic [17–19], recent presentations on international conferences [13–15] (all of which had their proceedings included in the Web of Science index in the last years) and a journal article [12], which is already accepted for publication.

This section references those works which (mostly loosely) relate to the subject or implicitly use results rationalization methods without any deeper considerations, mostly as intuitive solutions.

The following subsections cover:

- the usage and discussions of related methods of results rationalization and time series (subsection 2.2.1),
- result rationalization methods based on shifting windows (subsection 2.2.2) and probabilistic models (subsection 2.2.4),
- video segmentation methods splitting videos into sequences of strongly related frames (subsection 2.2.3),
- outlier detection in time series (subsection 2.2.5),

- video streams' and time sequences' continuity (subsection 2.2.6), as well as the properties of image similarity functions (subsection 2.2.7),
- related approaches in deep learning (subsection 2.2.8),
- a discussion of other related approaches (subsection 2.2.9).

2.2.1 Result rationalization

This subsection presents an overview of a big variety of works, which share one common feature: initial results of a processing or classification algorithm are altered in a second step. The additional step is supposed to improve the preliminary results on the basis of temporal properties of the analyzed underlying process. Those include especially known durations of events and change, which in typical multimedia recordings happens gradually over consecutive points in time.

The preliminary output of the first step often contradicts the natural expectations one might have about the underlying processes. For example, a heavily segmented output of a facial detector would indicate that a person is (dis-)appearing in the camera's sight much faster than humanely possible. A second step which ensures that the detected presences are rationally possible (e.g. by creating longer sequences of coherent classifications) can also be expected to improve the accuracy of the detection. Because the works presented in this section contain a step creating more rational predictions, they have been labeled *result rationalization* approaches.

The general idea exercised in this work corresponds to an algorithm presented in [69]. The authors of this article proposed composing a scene segmentation algorithm of simpler classifying algorithms, the results of which are later rationalized in terms of their temporal structure. Transition costs are assigned to scene changes and a most probable, least expensive scene segmentation is chosen.

Even though there are numerous classification methods for WCE video being proposed, virtually all of them work on a per-frame basis. Most algorithms acquire very high scores in terms of specificity and sensitivity, but many of the results remain questionable due to the small size and choice of their test data [31].

Most papers simply focus on classifying single images. Even if the significant similarity of frames in so called "events" [134] is acknowledged, the usual conclusion is that a single representative frame, usually the clearest shot, has to be chosen and classified. It should be mentioned that this approach does not teach the algorithm anything about handling the unclear frames in between.

Liu and Chen [101] have acquired a 10% better rate of recognizing text objects on frames when their temporal context in the video was considered. The proposed method has been shown to outperform a simple median filtering algorithm (with a window size of 3) by a factor of two.

The face-tracking algorithm introduced in [49] first detects faces as static features in separate frames of a video stream. Next, the face trajectories are smoothed using a Kalman filter and additional temporal criteria, what corrects occasional misclassifications. False face detections

are reduced six times after the second step is applied.

An additional step, which creates a consistent trajectory from detections of objects in consecutive frames, can be found in a number of vehicle tracking methods [72, 78]. It can be based both on domain specific approaches as well as common smoothing filters.

Smoothing itself is a method applied in various use cases, where an initial curve has to be transformed into a more probable form. For example in [106] an initial path of recognized hand movement is smoothed by a custom algorithm, which corrects possible mistakes in the initial hand position detections. Outlying elements are filtered out, while probably more correct positions are interpolated from the data.

A common tool for processing time series data with noise is the application of linear filters. They allow to remove noise from sequence data, therefore exposing an actual underlying process. Guidelines for applying filters in particular statistical applications (removing noise or periodical influences) have been provided in [3]. A noteworthy issue related to the use of filters is the endpoint problem (the necessary individual treatment of the values at the ends of sequences), which can be approached by fixed weights for all time series or by designing individual weights in each case. Also asymmetrical filters can be considered for the usage at the ends of sequences. It is noted though that they have worse properties than symmetrical ones, since they introduce phase shifts.

A wide discussion and analysis of the features and pitfalls of using methods based on linear filters has been provided in [115]. After listing the disadvantages of least square polynomial filters, it is noted that the binomial smoothing filter is more suitable for many uses. Especially – because it does not introduce a phase shift in a processed sequence. Data smoothing using low-pass filters has been also addressed by [77], where a number of filters have been discussed with a focus on their frequency response characteristics.

In a field closely related to video images, the matter of segmenting audio has been widely discussed in [1]. A number of methods have been proposed, which either already are or can be introduced in the subject of video segmentation. Significant attention has been dedicated to methods for preventing an overly segmented output, which directly relates also to single misclassifications in longer sequences. The matter of temporal coherence of the segmentation results is addressed and ways of modeling time distributions in various states (lengths of segments in the output) are discussed.

A similar idea has been used in processing classification sequences of audio recordings. In [130], after an algorithm recognizes the presence of voice in a recording, the initial results are smoothened. The applied smoothing method treats the initial sequence as a series of output signals in a hidden Markov model. A sequence of hidden states acquired with the Viterbi algorithm is significantly less fragmented and more accurate than initial results.

Turaga [151] presented numerous algorithms and analyses in the field of spatio-temporal pattern recognition in video, addressing multiple methods and presenting a detailed view of the whole field. A method of recognizing complex activities in videos has been defined. It composes

the activities from actions, which are defined over short sequences of consecutive frames. After an initial segmentation into actions, the action boundaries can be refined.

Approaches based on rationalizing results can also purposefully start with creating an initial result, which inevitably requires a further improvement. Grundmann et al. [57] proposed a novel spatio-temporal segmentation algorithm which starts with an over-segmented initial solution. Next, it incrementally creates a proper segmentation (here: segments are spatio-temporal shapes) by hierarchically joining pieces from an over-segmented initial result.

Video stabilization algorithms such as [148] rationalize the recorded movement of a camera. The smoothed, slower movement trajectory is expected to be the intentional motion, which allows to remove the jitter.

Stanisavljevic et al. [141] proposed an optical flow estimation algorithm which takes the temporal dynamics of the video stream into deeper consideration. The results acquired with post-filtering through five previous frames have shown to provide a significant improvement when estimating optical flow on a video with little textures. The flow fields tend to spread along the edges of moving objects and the actual movement can be exposed only after averaging a number of consecutive results.

2.2.2 Shifting windows

Cao et al. [29] has used a symmetrical sliding window of 5 frames to correct classification results (determining the presence of endoscopic instruments). The authors have only mentioned experimenting with windows of different sizes.

The actual influence of a shifting-window correction approach has been evaluated in [17], where it was shown that on an exemplary video over 96% of corrections were valid.

The continuity of video streams can be used to grow a shifting window and assign a common classification to sequences of similar frames [19]. Such an approach allows to accelerate the classification of the whole video in case of a computationally expensive algorithm classifying single frames.

Haji-Maghsoudi et al. [63] incorporated a post-processing step with a shifting window of size 5, where a majority vote determines the final classification of a frame (the segment of the gastrointestinal tract it represents). It's only briefly implied that this step improves the accuracy of their method.

In [119] and [120] an algorithm for detecting video frames with a circular content area is proposed. The results of the algorithm are post-processed to eliminate 'outliers' by assuring a uniform classification of long sequences of frames (100) and making the final frame classifications dependent on their neighbors.

Short sequences of 6 frames have been taken into consideration to detect changes in the video characteristics [52]. The windows whose left and right parts differed the most have been classified as events.

Another work using short windows is [22], where the recognition quality has been increased

after introducing a temporal filtering post-processing step. Statistics of detected hand postures are smoothed by using a formula analogous to an exponential moving average with a coefficient determined by the number w of frames in the window ($\alpha = \frac{1}{w}$).

The Temporal Sliding Window Detection method [38] has been proposed for finding short video segments containing actions of interest. The decision for a given sequence of frames was given by a SVM classifier. A fixed set of sizes of the shifting window (60, 80, 100 frames) has been used.

Figueira et al.[47] exploited the temporal continuity of pedestrians in videos to increase the performance of their detection. Bounding boxes of detected pedestrians in individual frames are classified with a classifier which provides ranked information. This information is in turn considered by a shifting time-window algorithm, which is parameterized by the maximal considered ranks, window size and confirmation threshold. A parameterization scheme has been proposed and guidelines provided for improving different measures of quality.

Shifting windows have also been used to identify local characteristics in time sequences and values standing out from their neighborhoods. For example, in [4] and [5] such an approach is used to identify when a car goes over bumps and obstacles. Neighborhoods of 10 seconds are taken, which is considered long enough to detect spikes of vertical acceleration, but also short enough to allow for real time processing.

2.2.3 Video segmentation

A broad overview of the scene segmentation methods and definitions available at its time has been presented in [56]. It covers not only methods of detecting *shot and scene* (or *scene and story*) boundaries but also proposes a method of evaluating them to express their quality.

Another general overview of approaches to temporal video segmentation has been provided in [86]. Different kinds of methods operating on full/compressed video streams and detecting gradual/instant transitions are discussed. Furthermore, the need to establish benchmark video sequences and evaluation criteria for video segmentation methods is emphasized.

The differences between scene/shot groupings in typical movies and those from endoscopic examinations have been outlined in [54]. Contrary to movies, whole endoscopic recordings can be considered to be single shots. The proposed scene segmentation algorithm is based on the presence of audio-cues from the doctor and detailed domain knowledge of the colon's anatomy – therefore it is irrelevant for our considerations.

In [168] a method has been proposed for the detection of smooth transitions between shots. In it, pairs of frames separated by a fixed interval are compared with each other to see, if significant change has happened between them. Change is considered to be gradual and different scenarios of its location relative to the compared frames are considered. Afterwards, the position of a camera break is narrowed down to an exact position in the video.

Shot boundary detection can be based on analyzing the variance transition of pixel intensities and the changes in the sequence of frame features (intensity pixel-wise difference, edge and

color histograms)[99]. The transitions considered in such an approach are either abrupt cuts or gradual changes, like fade-out or dissolve.

Lin and Zhang [96] introduced a multi-level temporal grouping: frames into shots, shots into scenes. Shots are grouped into scenes with an expanding window algorithm, which approximates the shot distance from their dominant color objects.

Video segmentation can also be used to create video summaries, to extract representative frames from each segment [150]. The segmentation can be performed by grouping frames into neighborhoods (temporal) in clusters (by similarity). The clusters are acquired by a non-negative matrix factorization of the similarity matrix of the frames.

The bottom-up method of grouping frames can also be used to segment an endoscopic video by organs [90]. First, the video is split into so called events. Next, the events are grouped by their correlation into solid, continuous ranges which represent organs.

When segmenting video material from movies or TV productions, grouping consecutive shots into semantically consistent scenes is the most challenging part of the task. One of the methods proposed for it is using a graph-based representation [140]. The visual distance between consecutive shots is one of the key factors in assigning them to a single scene. The decision about placing a scene boundary at a given location is based on a min-max cut of the similarity graph.

A number of related ideas has also been presented in [2]. Frames are grouped into scenes by clustering them on manifolds learned on inter-frame similarities. Optical Flow Divergence (OFD) and Normalized Cross-Correlation are introduced as measures of similarity. The OFD is interpreted as directly related to the movement of an endoscopic camera.

An algorithm for classifying whole video scenes has been outlined in [73]. The classification is based on motion information - optical flow or frame difference. For each class a HMM (hidden Markov model) is built, and the class with the maximal likelihood is returned as the classification. For one of the experiments the scenes are divided into shots using a simple histogram distance criterion. In [163] a similar approach is extended by including a possible concatenation of HMMs with a specified probability of switching from one semantic unit into another. [159] presented a more flexible approach, using a Hidden Conditional Random Field, which allows to create a single model describing all possible events.

2.2.4 Hidden Markov models

Hidden Markov models (HMMs) are probabilistic models, which consider a system of hidden states with a Markovian change rule and their visible observations. In the subject of this work, the hidden states can be considered to correspond to the ground truth, while the observations are classification results.

If a sufficient video data set is present, we can train such a HMM for rationalizing classification results. Afterwards, with a sequence of preliminary classifications two choices can be made. Either the Viterbi algorithm [48] for finding the most probable underlying sequence of ground truth states can be applied, or single values can be smoothed using a forward-backward

algorithm [139].

The Viterbi algorithm performs a sequential pass over the sequence of observed states to establish the most probable hidden state sequence. Approaches for the parallelization of the Viterbi algorithm have been presented in e.g.[45, 46]. Unfortunately they require a single sequential step for merging partial results and the parallelization factor is significantly limited by the number of hidden states.

Mackiewicz et al.[110] provided an overview of existing video segmentation approaches focused on segmenting WCE videos into parts that represent different organs. The tested approaches included classifying single frames as well as typical video segmentation methods (naive convergence, sliding window, HMM). The HMM model had a relatively simple structure, since only forward passes from one state to another are allowed (along the gastrointestinal tract.

A modified HMM of human actions based on features recognized in frames has been presented in [21]. The proposed HMM-MIO model (hidden Markov model for multiple, irregular observations) considers a variable number of features per frame as well as outliers and long tailed distributions. The proposed method is compared with single-frame classification and bag-of-features approaches, outperforming both significantly.

The Latent Dirichlet Markov Clustering method [171] assigns words to frames basing on their motion descriptor histograms. Each video is considered as a Markov chain over latent topics (actions). Later, a model is build that assigns co-occurring groups of frames to actions which are further grouped into video categories.

HMMs can be further extended into Hierarchical HMMs, which have been used e.g. for classifying patterns in video clips [166]. Their two-layer structure allows to model the inner variability within longer-lasting events. It is shown not only to outperform a regular HMM approach, but also k-means clustering and mining of co-occurring tuples.

Face recognition algorithms can be divided into still-image based and spatio-temporal ones, as discussed in [61]. One new still-image algorithm for recognizing faces in videos is proposed, which utilizes the intuitive fact that consecutive vectors from a video stream would lie on a smooth manifold. It is compared with a HMM-based algorithm, which utilizes the whole temporal sequence of features for recognition. The latter one has proven more efficient on downscaled images, as well as slightly better on longer shots of faces.

Markov sequences can also be noise-corrupted and methods have been developed to smoothen them [152]. A proposed algorithm for optimal smoothing of two-state Markov sequences (binary images) can be straightforwardly applied to the classification results of binary static features. It is worth to note, that if the noise at each position is independent, the Markov sequence with noise directly corresponds to a HMM, where the hidden states correspond to the actual, denoised sequence.

An overview of HMMs presented in [128] includes the topic of explicit state duration, which can be handled, but with a great cost related to the increase of both the computation time and the required memory.

2.2.5 Outlier detection

Finding misclassifications in a sequence of outputs from the previously defined OFA (One Frame Analyzed) algorithms is also directly related to finding outlying elements in time series if the assumptions about the video stream continuity are taken into consideration. Detected outliers can be treated as frames with a significantly higher probability of being misclassified.

A broad overview of outlier detection methods has been provided by Gupta et al. [59, 60]. Among others, the authors identify a number of approaches to identifying outliers in time series data, including detections based on forecasts, probabilistic models, clustering (and choosing outliers as the most distant to centroids), and a windowed analysis of subsequences. The detection of outliers has to consider the data trend drift and different kinds of outliers (additive–influencing a single point in the series, innovative–altering a number of subsequent values). Problems such as finding outlier points or subsequences in time series can be directly related with the scope of this dissertation.

Jagadish et al. [75] proposed a simple approach for finding deviants in time sequences by using an information theory approach and histogram representations. Those elements, whose removal allows to create a closer representation of the sequence even with fewer histogram bins are considered to be deviants. Later, another algorithm is presented for grouping deviants and finding whole deviant intervals.

Numerous other methods listed by Gupta involve the usage of prediction models which compare a single value with its prediction based on its neighborhood (one-sided or two-sided). Prediction models can include the median of the values in the neighborhood [8], their average [66] or more complex approaches (nearest cluster, linear regression, multilayer perceptron [66]). These methods actually correspond directly to the shifting window approach presented in subsection 2.2.2 with the neighborhood being equivalent to the shifting windows.

Another approach to detecting outliers in time sequences, proposed by Ma and Perkins [108], is closely related to the shifting window approaches. The authors transform the time series into a sequence of all fixed-length subsequences of the series. The resulting vectors are classified using a one-class SVM. A point whose corresponding vectors are identified as outliers for a number of subsequence lengths is reported as an outlier in the time sequence.

2.2.6 Continuity

Analyzing the generalized continuity of the video stream and the classification function leads to the question of defining the corresponding differences (metrics, similarity functions).

In terms of image differences multiple works have addressed the subject in various usages.

Zhao and Meng [173, 174] proposed algorithms for finding frames with sudden changes in WCE videos. They are based on a distance function using Euclidean distances between color, texture, and shape feature vectors. Since the movement of a WCE capsule in the gastrointestinal tract creates a recording with relatively little change, the appearance of any lesions, bleedings

or other abnormalities results in a faster pace of change in the video. This allows to identify the key frames for a further summarization of the video.

Of course, the feature/image spaces as well as the ultimate classification value spaces are not really continuous in terms of the according analytical definitions. Nevertheless, the concept of generalized continuity on discrete sets has been already deeply researched and formalized in works such as [27], which provides new definitions and solutions for the generalization of the intermediate value theorem.

An earlier and groundbreaking formulation was the theory of rough sets whose foundations were laid by Pawlak in his works [85]. An aspect of this work were the papers regarding rough calculus [124, 125] - defining rough continuity, Darboux property, and even differentiability. Those ideas have later been extended and developed in various works such as [160] discussing rough derivatives or [64], where the rough sets theory has been applied in image classification tasks. Especially worth noting is the ability of rough set methods to handle ambiguities in interim regions.

Furthermore, works such as [146] use the continuity of videos to support tracking objects by predicting their position. This allows to limit the region of search for other object detection algorithms.

2.2.7 Image similarity metrics

While various works presented in the other sections focus on arbitrary metric functions for images, there are also some works discussing image distances in a more general and structured way.

A field where image similarity metrics are used widely is image retrieval. The metrics are used to compare and assign images to particular classes. In [44] a simple function based on the compared images histograms and the Dynamic Time Warp method is used.

Wang et al. [158] analyzed the performance of multiple metrics on color histograms, LBP and Gabor feature vectors, and an edge histogram descriptor. Their proposed fractional distance metric has been superior to other ones in the majority of cases.

Image similarity can be used in image registration, when two distinct images need to be aligned with each other. Li and Stevenson [92] introduced a modified Hausdorff distance based on curves in images, which improved other approaches such as a straightforward pixel-wise similarity and implicit similarity [83].

Comparing multiple images with each other, as might be the case when videos are considered, can lead to significant computational costs. The properties of metric functions can be utilized to develop approximate solutions, e.g. using the correlation of distances from two considered images to other ones as a measure of their similarity [74]. Similarly, the distances between learned SVM boundaries and images in a database can be used in image retrieval as a measure of similarity with a given query [58]. Such an approach has shown to be more effective than using the Euclidean distance from the query.

Rubner et al. [137] presented a series of histogram-comparison methods, especially focusing on the Earth Mover Distance (EMD). Other methods, such as L^p norms, histogram intersection and information theory-based approaches are discussed with their potential applications. It is noted that EMD allows to account for minor shifts in color schemes or luminance. The authors also compare the efficiency of using image signatures (describing an image by splitting pixels into similar subsets and creating a vector of the mean values and cardinalities of those subsets)

Different metric distance functions can be related to specific understandings of mean vectors in a multi-dimensional space [170]. Metrics corresponding to a generalized harmonic mean turn out to be more robust to single outliers and show the highest statistical correspondence when applied to comparing images with their distorted versions. The observations of the improper application of simple distance metric for feature vectors (which can have very different distributions of values in each dimension) lead to the proposition of training a metric function as a sum of weak classifiers, which performs better than simple metrics.

A wide survey of metric learning methods has been presented in [9]. These are introduced due to the difficulty of manually creating specially tailored metrics for particular tasks. Particular attention is dedicated to the Mahalanobis distance metric-learning framework. Other methods, such as nonlinear metric learning, similarity learning, as well as local metric learning are introduced, altogether with methods for structured data. Most of these methods require the preparation of a training set for the learning process.

An easier method for training similarity metrics is to just provide similar/dissimilar information for pairs of images. Such a method has been proposed for medical image retrieval by Cai et al. [28].

2.2.8 Deep Learning

Another approach to processing video, which has gathered a lot of attention in the last years, is the application of deep neural networks. Two main architectures which need to be considered are convolutional and recurrent neural networks (CNNs and RNNs). A review of the origins, achievements and perspectives of deep learning, focused on the two aforementioned types of networks, has been presented in [89].

CNNs are currently the basis of numerous cutting-edge methods related to image classification. They usually consist of two parts: the feature-extracting convolutional layers and the fully connected classifying ones. The convolutional layers are usually interleaved with pooling layers, which are responsible for reducing the dimensionality (albeit some works acquire better results without them [100]). They have shown to be a great improvement in terms of classifying images and have also been applied to video frames on numerous occasions. Numerous methods keep emerging, striving to improve the accuracy and efficiency of training and image classification. Improvements are often related to new network architectures, as e.g. residual learning [65] with connections skipping layers, or networks like Inception-v4 and Inception-Resnet [145] which are constructed as a composition of uniform modules (the latter also with residual connections). Also

different approaches to training can be noted, such as [97] where the well-classified examples are down-weighted in the loss function to focus the training on more difficult cases.

The presence of a given kind of object is an example of a static feature. It can be detected in a single image, but can also be expected to last across multiple frames of a video. For such tasks deep neural network object detectors can be utilized. Two major trends are the more accurate two-stage detectors (which first detect bounding-box candidates for further classification, e.g. Faster R-CNN [133] or R-FCN [36]) and the faster single-stage networks (e.g. YOLO [132], YOLO9000 [131] or SSD [103]).

RNNs are networks which introduce recurrent connections, i.e. consecutive classifications are related through an internal "state" of the network. This allows to model temporal relations, like those in video sequences. Architectures based on LSTM cells [67] have allowed to surpass issues of recurrent networks, such as vanishing gradients and have proven effective in multiple fields, including video classification [76] and summarization [114].

The structure of the memory cells in RNNs implies a one-directional flow of time, which can be related to the online processing model. Nevertheless, offline models can also be pointed out. They utilize the availability of the full input sequence in bi-directional structures (e.g. the saliency-based classification [126], described below) or explicitly separate a subsequence of the input for classification using other methods (e.g. [35], where a simple statistic is used to identify an ongoing gesture in photodiode readings, whose corresponding subsequence of readings will be classified using the RNN only upon its completion).

An approach which introduces an additional LSTM layer on top of a SSD object detector [107] corresponds to the ideas presented in this dissertation. An additional layer adding the temporal context to object detection results from consecutive frames has allowed the authors to track objects even on frames where the underlying detector failed to find them. Also in [105] initial predictions of face saliency acquired from a CNN network are passed through a LSTM network to improve the saliency model over a video sequence.

The authors of [167] used a convolutional-LSTM network for crowd counting in video. In their analysis they underline the availability of the temporal information, which remains unused by traditional CNN approaches. Furthermore, they propose a bidirectional approach, which allows for analyzing a frame on the basis of long ranges of frames in the video in both directions (preceding and succeeding frames).

The results of [6] indicate that a CNN architecture can be proposed for sequence processing which outperforms RNN/LSTM based approaches and ensures better long-term memory. The proposed architecture has been named *Temporal Convolution Network* and consists of a series of dilated convolving layers, which are able to cover the whole sequence in the receptive field.

Sun et al. [144] evaluated a number of methods for labeling of whole videos. Their considered approaches included a global vote (returning the class with the most votes), Temporal Feature Pooling (convolutional layers for each frame, connected with pooling and fully connected layers which return a classification of the whole video), a deep LSTM network, and a 3D convolutional

network trained on whole videos. The first three methods acquired comparable accuracy results (57.8%, 57.0%, 61.5%) using a pretrained CNN. The training of the 3D CNN was unsuccessful due to time and training data requirements.

A network architecture for classifying sequences by weighting element in the sequence using their saliency has been proposed in [126]. Tested applications included speech recognition, natural language sentiment evaluation and video labeling. This approach has been shown to outperform other RNNs in most cases, since it allows to focus the classification of a sequence on its informative parts and ignore noisy ones. The training set used for classifying videos consisted of 4659 videos with an average duration of 80 seconds (over 100 hours of video).

From the perspective of this work, deep neural networks can be considered threefold. First, typical convolutional networks are representatives of the OFA approach. For each frame of the input video they provide an independent classification, without considering the frame's neighborhood. Secondly, methods which combine convolutional feature extraction and recurrent layers are considered self-contained methods and are therefore out of the scope of this dissertation (which is focused on improving OFA methods). Finally, the last group are networks or approaches which build upon the output of preexisting image classifiers (possibly but not necessarily also deep neural networks). Those methods are applicable in similar scenarios as the FSA approach proposed in this work. Therefore, only this last group of deep learning solutions has been considered in the solution overview in subsection 2.2.10.

2.2.9 Other approaches

Cox and Snell included a section regarding binary time series in their book on binary data [32]. Because binary static features are considered, the OFA result outputs are binary time series, and therefore the proposed methods of analysis can be applied. Among them are: treating the series as a realization of a Markov process for likelihood estimation, treating the binary values as a function of an underlying Gaussian variable.

Kedem and Fokianos [80, 81, 82] presented numerous regression models of binary and categorical time series. The authors introduced the logistic regression for time series, presented conditions guaranteeing the applicability of a probabilistic approach in time series, and discussed appropriate "goodness of fit" measures for binary data. Categorical variables are divided into ordered and not ordered ones and appropriate regression models have been proposed for them – mostly derived as generalizations from the binary models.

A typical approach for the general task of smoothing a time sequence is to filter out higher frequencies in a spectral representation of the sequence. Since consecutive values are usually correlated (contrary to noise), they are mostly represented by lower frequencies. Such an approach was the base for works such as e.g. [24, 25, 41].

Methods developed for single images can often be generalized into the three dimensional space, by treating the time axis the same way as x and y coordinates. Sukthankar and Hebert [143] classified efficient volumetric features based on integral videos. Events are sought in the video in

spatio-temporal boxes, classified by a cascade of simple classifiers based on thresholding single features.

Methods related to the segmentation of 3D images [42] can be also considered for detecting the spatio-temporal evolutions of objects. Such an image can be conceptually equivalent to a video recording – even more if the third dimension corresponds to consecutive slices which were acquired sequentially. Applying such an approach requires at the same time that the pixel neighborhoods are considered carefully. Especially, it may require to adjust the spatial coordinates of neighborhoods in the temporal direction using the optical flow of the video [57].

Training OFA algorithms on single images which often are acquired from frames of the highest quality results in a relatively low stability of the output classification sequence. To mitigate this effect a stability score can be included during cross-validation in the progress of training an algorithm which classifies single frames [34]. The video used to evaluate the stability does not need to be fully annotated or come from the same training set. Tests have shown that adding the stability score when training a classifier improved its stability, with a minimal loss in accuracy.

The authors of [40] have used the similarity of neighboring frames in a video for inpainting, i.e. filling the space in an image after removing an object from the frame. After registering the source frames (adjusting the images to account for movement of the camera or in the view), a covered or removed object can be restored by considering the relevant data in source frames and a smoothness measure, which ensures acquiring a coherent shape.

A number of aforementioned fields related to processing video streams, such as video summarization, indexing and retrieval, can be grouped together in a broader field called Video (Object) Mining [162], which is conceptually related to data mining. The temporal aspect of videos is noted to introduce significant computational and memory requirements in this field, albeit often with a limited use. Focusing on video objects (objects visible throughout multiple frames) allows to introduce the temporal aspect and change the focus of video analysis from the typical global/shot/frame levels.

Gama [53] presented a survey on learning from data streams, which presented methods processing approach very long streams of data online. The presented distinction between processing data in databases and in streams places the methods considered in this dissertation closer to the former ones. This is determined by factors such as data access and number of passes (databases: random+multiple, streams: sequential+single) and result accuracy (databases: accurate results, streams: approximate results, with a lower precision for older data). If online processing algorithms were considered, dedicated tools developed for online learning from data streams [11, 118] can be used to implement the methods in this field.

2.2.10 Related methods summary

A categorization of methods emerging from the presented overview has been summarized in Table 2.2. A number of aspects important from the point of view of the considered application have been included. The presented types of methods are not fully separable classes. For example,

small linear filters are de facto shifting windows and the detection of outliers can be performed by a number of other listed methods. The approaches listed in the table represent the base concepts of the discussed works and help with describing their applicability and limitations.

The analysis scope of a method is the range of frames, which provides the information for improving the classification of a single frame. It can be either a small local neighborhood (as low as a couple of frames) or even the whole video sequence, when the whole course of changes in the video is predicted to improve individual classifications. Depending on the method, the scope of interest can be also limited to a one-sided neighborhood (only frames preceding a considered one), which allows for online processing. Methods which analyze a small number of frames after the considered one can provide online classifications only with an inevitable delay.

Another differentiating factor is the amount of training data required for a given method to ensure an improvement of classification results. Some of the considered methods can start improving the classification results "out of the box" – with a high confidence but possibly not at an optimal level. On the other hand, the application of a RNN would require the preparation of an extensive training dataset. If the network was also tasked with classifying the images, the gain of using a pre-trained OFA method as a black box would be lost and the amount of necessary data would increase dramatically.

We also consider the possibility of parallelizing the analyzed methods, especially for distributed computation. Since the considered target usage involves processing of very long archived video recordings, methods allowing for distributed processing are preferred. In the *Parallelization options* column only such options are considered. Local approaches, such as loop parallelization on multi-core CPUs or processing appropriately structured data on GPU units are almost always possible in digital image processing and are therefore not included in the consideration.

The *Comment* column contains additional remarks regarding particular methods. The Shifting Window approach requires to account for the size of scenes when establishing the size of windows (smaller ones might lose information, larger ones might consider irrelevant data and perform worse). HMMs and RNNs have issues related to the cost of execution or training. Outlier detection methods are best suited for detecting the additive outliers (single points). In a dynamically changing sequence (e.g. classifications in distorted video), the misclassifications are not as prominent. Video Segmentation approaches often start with finding the easily determined shot boundaries. Finding scenes, which are identified mostly semantically, in single-shot recordings is one of the most difficult aspects in this field. Filters are used in signal processing to remove unwanted ranges from signals in the frequency domain. They are applicable if misclassifications are considered in terms of noise in the classification sequence, which is mostly the case. This approach is strongly connected with the shifting windows if only linear filters were considered. Nevertheless, some filters (e.g. Kalman filter) are applied in a manner which corresponds to the online approach and considers the past part of all frames.

Table 2.2 – Comparison of methods.

Method	Analysis scope	Training data requirements	Parallelization options (non-local)	Processing model	Comment
Shifting windows	Local neighborhood (seconds)	Low	window-wise	Offline Online with delay	Window size related to scene size
Hidden Markov models	Whole recording	Medium	limited	Offline	Inefficient parallelization
Recurrent neural networks	Past part of recording	High	limited	Online	Expensive dataset+training
Outlier detection	Whole recording or local	Low	method-dependent	Offline Online with delay	Additive/innovative distinction
Video segmentation	Whole recording or local	Low	limited	Offline Online with a significant delay	Additional difficulty without shot boundaries
Filtering	(determined by the receptive field of the filter)	Low	from trivial to limited (determined by the receptive field)	Offline Online with delay	Numerous approaches in signal processing
Domain specific	Local – Global	Low – High	method-dependent	method-dependent	

2.3 Vision and general proposition

The discussed methods for addressing the subject of rationalizing OFA classification results have been divided into a number of approaches in the previous section. As it has been noted, the shifting window approach turns out to be often used with nothing more than an intuitive justification. Only some works imply or vaguely confirm that it contributes to improving the quality of classifications. In this work we are going to elaborate more about the reasoning behind this approach and also propose and evaluate a structured way of extending it and adjusting its parameters.

The research problem presented in the previous sections, can be summarized as follows: We want to efficiently and universally improve existing OFA algorithms by considering the temporal structure of their classification sequence. It will be addressed in this work by proposing a general method which improves preliminary OFA classifications by processing them with the shifting time-window approach. The improved algorithms will consist of an underlying OFA algorithm and a function which transforms the OFA output for a sequence of frames. Its aim is to ensure a level of output continuity, which results in an improved classification quality. This new kind of algorithms will further be called Frame Sequence Analyzing (FSA) algorithms.

The methods turning OFA algorithms into their corresponding FSA versions will be the main consideration of this work. The performance of the resulting new FSA algorithms will be evaluated in terms of improvements in:

- sensitivity and specificity, and
- output stability

in comparison to their OFA counterparts. Concrete quality metrics for those criteria will be presented later, in section 4.7.

According to the definition of the FSA approach we can define two main phases of processing:

1. The OFA phase - where preliminary classification results are acquired from the base algorithm.
2. The **result rationalization** phase (FSA step) - where the preliminary sequence of results is processed into a more plausible version.

2.4 Thesis statements

The presented analysis of related work gives an intuitive understanding of how applying FSA methods can improve the performance of image classification algorithms in videos. The only preconditions are the availability of an OFA method for a given problem and the continuity of the classified video. The results of experiments using such an approach implicitly are promising and give reason to evaluate this subject further.

Figure 2.3 – Illustration of thesis statements. *Errors* - error rate

In this dissertation we will propose the FSA approach as an original method which addresses all of the presented considerations. We will analyze two FSA variants – iFSA and fFSA– and multiple parameter sets for them. Those methods, the data they operate on and video continuity will be formally defined and evaluated in the following chapters.

The proposed methods can be evaluated only in comparison with the underlying OFA problem and its data. Our assumptions are that the FSA method will both provide better results than OFA and be more robust to changes in the input data quality.

After the analysis and discussions earlier in this chapter, we preliminarily assume that the proposed FSA method will perform better then their corresponding OFA methods in two ways[2]:

1. For an optimal window width the FSA methods will perform better in terms of result accuracy than their corresponding OFA ones on the same video stream.
2. In terms of accuracy, the FSA methods will perform on a distorted version of a video stream just as well as their corresponding OFA ones on the original stream.

The sense of the two thesis statements has been illustrated in Figure 2.3. The first thesis statement concerns the OFA and FSA methods being applied on the same version of a video stream. In such a case, we claim that the FSA method is able to lower the error rate by 20 percent. The second thesis statement addresses the situation in which the FSA method is

[2]Exact phrasing from the initial application (in Polish):

1. Metoda FSA daje wyniki porównywalne do OFA przy większym stopniu zakłócenia strumienia.
2. Metoda FSA daje lepsze wyniki niż OFA w sensie wiarygodności, obniżając liczbę błędnych wskazań dla optymalnej szerokości okna.

classifying a video stream whose quality has been decreased. In such a case, we expect it to uphold the quality of the OFA method on the original recording.

In the next chapters we will define a number of terms required to rephrase those statements in a more precise manner. Chapter 3 will discuss the definition of a continuous video stream as well as the underlying probabilities and optimal window sizes. In Chapter 4 we will define four quality measures (FPR, FNR, IBR, MBR) for evaluating the quality of classifications. The first three of them are combined to express the classification quality. The MBR measure has been introduced as a symmetrical measure for IBR, and is discussed in the result analysis. It is important to note, that those measures express the level of mistakes of an algorithm, e.g. an accuracy of 90% can be associated with FPR and FNR being equal 10%. The goal of introducing the FSA method is therefore to decrease the values of those measures.

Next, in Chapter 5 we will define a way of expressing the intensity of distortions in a video stream. It is not possible to measure the loss of image quality in a perfectly objective way. Therefore this work bases its definition on the given distortion type's intensity scaled from 0 to 100%, where the latter marks the value for which the images start becoming incomprehensible.

After introducing all of those preliminaries, we will be able to rephrase the thesis' statements in a more precise form, holding for both of the FSA variants:

1. For an optimal window width the FSA method performs better than OFA in terms of result accuracy (reducing the number of classification mistakes on average by 20%):

$$\mathrm{FPR}_{\mathrm{FSA}} \leq 80\%\mathrm{FPR}_{\mathrm{OFA}} \tag{2.1}$$

$$\mathrm{FNR}_{\mathrm{FSA}} \leq 80\%\mathrm{FNR}_{\mathrm{OFA}} \tag{2.2}$$

   and result stability (reducing the number of scene segmentation mistakes on average by 20%):

$$\mathrm{IBR}_{\mathrm{FSA}} \leq 80\%\mathrm{IBR}_{\mathrm{OFA}}. \tag{2.3}$$

2. Even on streams distorted by 10 intensity percentage points, the FSA methods will perform just as well as OFA on the corresponding less distorted stream in terms of all quality measures.

3 A formal ground for the FSA approach

This chapter presents a more formal and in-depth approach to the considered problem of classifying frames in video streams. First, a general definition of the classification function is proposed to create an analytical approach. To do this, we define the classification task as a numerical function in the context of other related functions and define video continuity in terms of their properties.

Then, an analysis of the properties of video data which is considered to be continuous is presented. This gives a formal ground for combining the reasoning for temporally close frames.

Next, a short preliminary experiment is described, which considered another approach to improving OFA classifications – improving classification time at the cost of worse accuracy.

Finally, the motivations for using shifting time windows are discussed, what involves computing the correctness probability for a simple decision function which improves classification results. Since already such a simple algorithm is promising an improvement, we find it justified to perform tests on it and its generalizations.

3.1 Classification in video sequences

3.1.1 Image classification

Let us define an image classifying function (OFA algorithm) as a function

$$c : \Theta \to \mathcal{C}$$

where:

- Θ is the set of all possible images for a given problem,
- $\mathcal{C}$ is the set of all possible classifications for the given problem.

The value of $\mathcal{C}$ is defined by the considered classification problem. For example:

- for binary static properties $\mathcal{C} = \{0, 1\}$,
- for binary static properties with a certainty or probability level $\mathcal{C} = [0; 1]$,
- for numerical properties $\mathcal{C} \subseteq \mathbb{Z}$ or $\mathcal{C} \subseteq \mathbb{N}$,
- for k binary labels $\mathcal{C} = \{0, 1\}^k$:

  - if always only exactly one label per classification is allowed [3] then this case is limited to: $\mathcal{C} \subseteq \{0,1\}^k$, such that $\mathcal{C} = \{i^*\}_{i\in[1..k]}$ where $i^* = (\underbrace{0,\ldots,0}_{i-1},1,\underbrace{0,\ldots,0}_{k-i})$,

- for k labels with a certainty or probability level $\mathcal{C} = [0;1]^k$:
  - if the classification is a probability distribution of all possible labels, then this case is limited to a $\mathcal{C} \subseteq [0;1]^k$ such that $\forall_{C\in\mathcal{C}} \sum_{c\in C} c = 1$ (every classification has to sum up to 1).

Due to the noise and redundancy of information in the highly self-correlated images in video streams, the classification function is actually an implicit or explicit composition $c = \kappa \circ \phi$ of two functions:

- ϕ - the feature vector function (FVF),
- κ - the classifying function.

These two functions need to be distinguished as their application and ways of obtaining them are significantly different.

The purpose of the FVF is to decrease the input data dimensionality - from the full image to a limited set of so called **features**. The multidimensional feature vectors, on which the actual training is performed, are expected to describe some specific characteristics of the classified frames. These are acquired from the color distributions, recognized edges, local binary patterns, and various other statistics of the images. Obtaining a proper FVF is a matter of domain specific knowledge or experimental selection from a larger set of candidate features. This can involve, for example, the usage of color features for finding bright red bleedings or edge density detection when searching for particular textures. In the case of convolutional deep neural networks, FVFs are essentially learned (as weights in the convolutional layers) from raw training data. The space of all considered feature vectors is denoted as $\mathbb{F}$. It is acquired as a transformation of all images in Θ using the FVF ϕ (i.e. $\mathbb{F} = \phi(\Theta) = \{\phi(\theta) : \theta \in \Theta\}$). The multidimensional vectors acquired from any kind of FVF are later used as the basis for the classification of their underlying images.

Continuing, the classifying function κ can be provided as one of the established classification methods: a neural network, an SVM classifier, a decision tree, a cascading classifier of simple features [154, 155] etc. Such a function κ is acquired by applying a training algorithm of the corresponding method.

The whole classification process is symbolically presented in Figure 3.1. Among OFA algorithms a significant number (e.g. [84, 112, 113]) can be pointed out which share many common characteristics, including this general outline of the algorithm. In general, the following steps are performed with a given training set $U \subset \Theta$ of images with known classifications:

1. $\phi(U)$ is computed – the images are transformed into feature vectors, according to a specific algorithm (the FVF ϕ).

[3] This is equivalent to $\mathcal{C} = [1..k]$, but can be more convenient in notation and computation for a composition of multi-feature classifiers

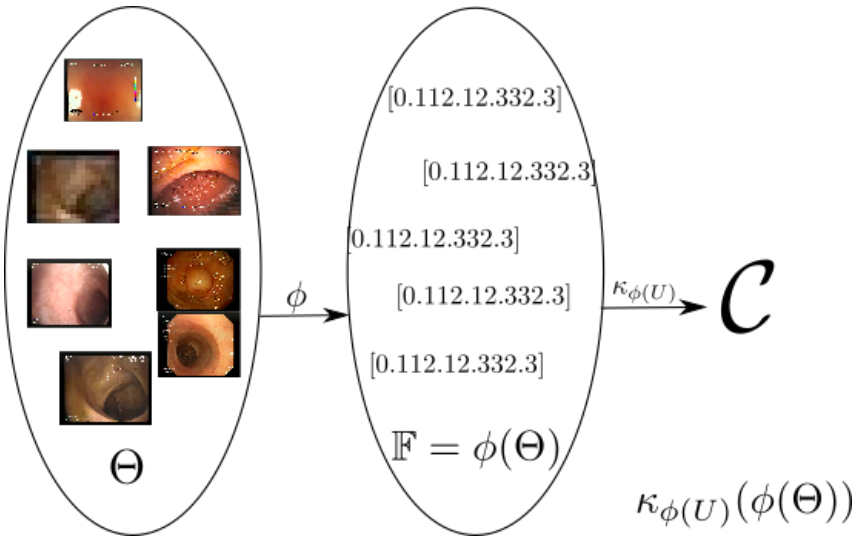


Figure 3.1 – The feature vector and classifying functions and their domains/codomains.

2. $\kappa_{\phi(U)}$ is learned - the feature vector-classification pairs are used for training a classifier. Established Machine Learning methods can be used interchangeably and evaluated in terms of accuracy.
3. The image classification function is created as the composition of the feature vector function and the classifier function $\kappa_{\phi(U)} \circ \phi$ – with which new incoming images can be classified.

Table 3.1 presents the classification functions considered in this dissertation in terms of the definitions above. As one can see, the split of a classifier c into ϕ and κ might be ambiguous for some functions. For a classifier impossible or difficult to decompose, we can also assume ϕ to be an identity function and $\kappa = c$.

Next, we will continue by discussing a generalization of continuity for the (in fact discrete) classification function applied to discrete video streams. We will take the whole classifying function into consideration, keeping in mind that it is in fact a composition of the aforementioned two functions.

3.1.2 Properties of continuity

In most cases, real-life processes recorded on videos are of a continuous nature. In this and the next sections we will analyze what properties of this continuity transgress into the video domain and how they can be, explicitly or implicitly, exploited when sequences of consecutive video frames are classified.

The problem of continuity has already been mentioned in the previous sections a number of times. So far we have separately mentioned the continuity of the observed reality, the video stream and the classifying functions. It has also been pointed out that the space of images that we are considering (and therefore also all other spaces acquired by transforming it with any function) is in fact finite. Furthermore, also the time dimension is discrete, though not necessarily

Table 3.1 – Classifying functions

Classification	**Silhouette detection**	**Face detection**	**Traffic light recognition**
Domain, Θ	1680 x 1050 24bit	800 x 600 24bit	640 x 480 24bit
Codomain, $\mathcal{C}$	True/False	True/False	True/False
Features, $\mathbb{F}$	Haar-like	Haar-like	thresholded channels
κ	Cascade model trained for silhouette recognition	Cascade model trained for face recognition	Custom function
$c =$	**true** if any silhouette visible, **false** otherwise	**true** if any face visible, **false** otherwise	**true** if a green traffic light visible, **false** otherwise

finite. Depending on whether full videos or live video streams are considered, without loss of generality it can be represented as $\mathcal{T} \subseteq \mathbb{N}$.

Nevertheless, both the image dimension and the time dimension are in fact projections of continuous spaces onto their discrete counterparts in the digital video stream. And the perceived continuity of the recorded processes can be lost only due to a frame rate insufficient for the pace of change in view.

In the field of continuous functions the regularity and pace of change are expressed in terms of continuity and derivatives. For our considered functions we need to adapt those terms to account for the discreteness and finiteness.

Discrete continuity

Two of the most general types of continuity of a function $f : X \to Y$ defined on metric spaces X and Y include regular *continuity*, as per Cauchy's definition:

$$\forall_{x \in X} \forall_{\epsilon > 0} \exists_{\delta > 0} \forall_{x_2 \in X} : d_X(x, x_2) < \delta \Rightarrow d_Y(f(x), f(x_2)) < \epsilon$$

and its stronger form, *uniform continuity*:

$$\forall_{\epsilon > 0} \exists_{\delta > 0} \forall_{x \in X} \forall_{x_2 \in X} : d_X(x, x_2) < \delta \Rightarrow d_Y(f(x), f(x_2)) < \epsilon$$

where d_X and d_Y are metrics of the spaces X and Y, respectively. Uniform continuity implies continuity, but there is no implication in the other direction.

If X is a set of isolated points, in both cases the choice of a value of δ is trivial and allows to prove any function to be continuous. Namely, such a value of δ has to be chosen, which isolates

each considered point in X to a one-element neighborhood.

Another way of defining the continuity condition is directly limiting the differences of function f values with a continuous function. Below we will first describe such limiting functions and then prove that the proposed definition is equivalent with the definition of uniform continuity introduced above.

Definition 1. As the limiting function we will take such a $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, which fulfills the following conditions:

1. $g(0) = 0$,
2. g is monotonically increasing and continuous .

A function fulfilling those criteria will further be called a *continuous limiting function*.

After choosing a function g which is a continuous limiting function we can now define how other functions can be limited by it.

Definition 2. A function $f : X \to Y$ is *g-continuous* if g is a continuous limiting function and if:

$$\forall_{x,x_2 \in X} : d_Y(f(x), f(x_2)) \leq g(d_X(x, x_2)). \tag{3.1}$$

Theorem 1. If $f : X \to Y$ is g-continuous and X and Y are metric spaces, then f is uniformly continuous.

Proof. For any given $\epsilon > 0$ let $\delta_\epsilon > 0$ be any value such that $\forall_{t<\delta_\epsilon} g(t) < \epsilon$. Since g is continuous (in particular, in $t = 0$), such a value always exists.

Continuing, we acquire:

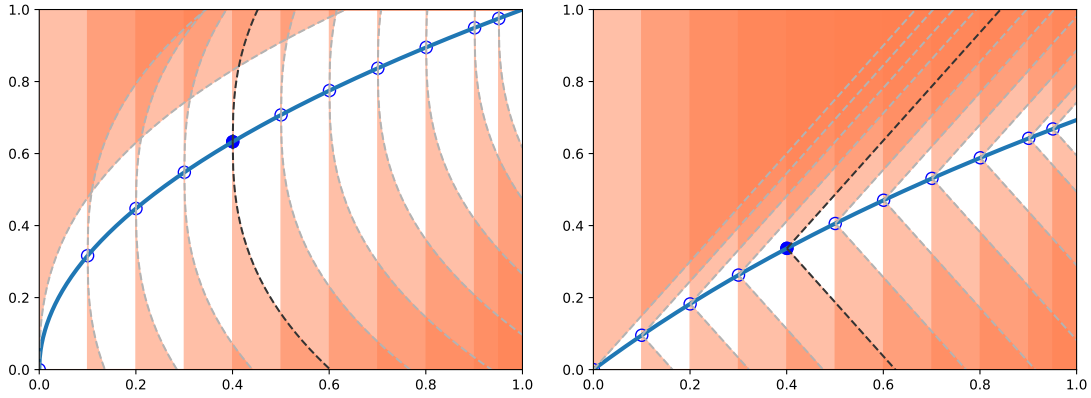
$$\forall_{\epsilon>0} \exists_{\delta=\delta_\epsilon} \forall_{x \in X} \forall_{x_2 \in X} : d_X(x, x_2) < \delta \Rightarrow \epsilon > g(d_X(x, x_2)) \geq d_Y(f(x), f(x_2)).$$

Therefore f is uniformly continuous. □

Having defined continuity in this way, we can account for the "undefined" periods between the values given at discrete time points and still set some limits on the amount of change between known values. A graphical illustration of the application of a given continuous function g and the corresponding g continuity limit has been presented in Figure 3.2.

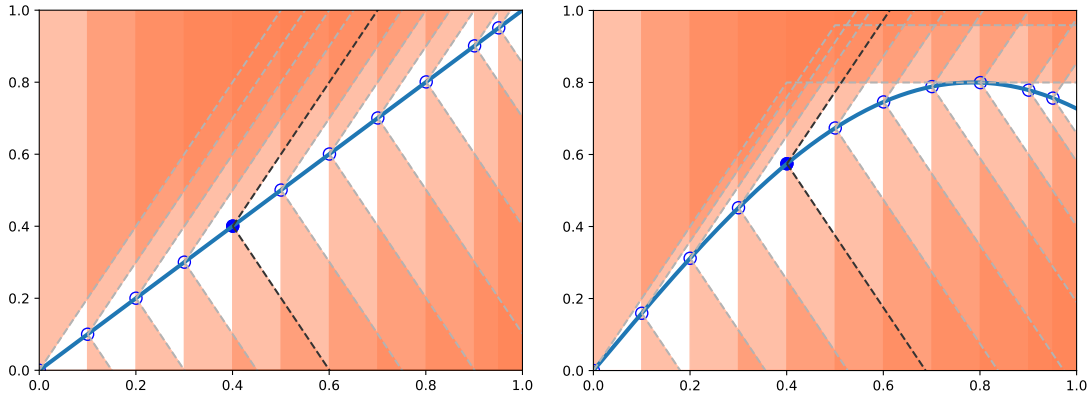
A new question about the choice of the g function arises. The choice of a sufficiently steep g would result in accepting all considered discrete functions as g-continuous. In such a case the check for g-continuity would be pointless.

It is worth noting that the values of a g-continuous function f in Equation 3.1 are limited not only by the course of g from the closest points. Also limitations from different positions overlap, as it is shown in Figure 3.2. Therefore, to properly enforce the limits on the pace of change of function f we need to ensure that it is changing slower than the limit function g at any point.



(a) $f(x) = \sqrt{x}$
$g(\delta) = 1.2 \cdot \delta^{0.4}$

(b) $f(x) = \log(x+1)$
$g(\delta) = 1.5 \cdot \delta$



(c) $f(x) = x$
$g(\delta) = 2\delta$

(d) $f(x) = \sin(2x) \cdot 0.8$
$g(\delta) = \begin{cases} 2x & \text{for } x \leq 0.4 \\ 0.8 & \text{for } x > 0.4 \end{cases}$

Figure 3.2 – Explanation of the g-continuity definition. The continuous (blue lines) and discrete (round markers) f functions are all compared to the limits given by g functions (dashed lines).

This leads to the conclusion that g-continuous functions need to be at most linear, since the change of the functions value has a fixed upper bound.

This observation also limits the choice of reasonable *continuous limiting functions*, since every superlinear function will anyway result in a linear limit on f. Therefore, the choice of a linear function g allows for the maximal possible rate of change. This leads us to another important kind of continuity:

Definition 3. We call a function $f : X \to Y$ *Lipschitz continuous* with the Lipschitz constant L if

$$\forall_{x_1,x_2 \in \mathbb{X}} d_Y(f(x_1), f(x_2)) \leq L \cdot d_X(x_1, x_2). \tag{3.2}$$

The above definition is equivalent to bounding the absolute value of the derivative of f with L. In terms of video streams with discrete time this can be understood as putting a limit on

the incremental change between consecutive frames in the video by setting the constant L to an arbitrary value. Although, for such a limit to make sense, the value L should be sufficiently low, so that it does not allow for all possible value changes in the discrete and finite space of the considered function's values.

The discussion presented above warrants a question about the use of other than linear continuous limiting functions in describing a continuous process. Let us consider a continuous limiting function g defined as in Figure 3.2d:

$$g(x) = \begin{cases} 2 \cdot x & \text{for } x \leq 0.4 \\ 0.8 & \text{for } x > 0.4 \end{cases}$$

The chosen function g expresses both the pace of change allowed between given frames and also limits the maximal extent of the change. This limit reflects not just the finite image space, but also physical and domain-specific restrictions. Those can be for example positions of to-be-identified objects in the field of view, whose movement is limited by the dimensions of the observed area.

Further on we will focus on the Lipschitz continuity, which is the minimal requirement for the applicability of the following reasoning.

3.1.3 Continuity of videos and classification sequences

In this subsection, we will discuss how the continuity of a real-life process is preserved in its video recording, which is a sequence of discrete frames in discrete time. We also have to consider the inaccuracy introduced in the recording.

These considerations are significant, because they allow us to formally express our intuitive perception of real-life video recordings. They explain why we actually can expect consecutive frames of the video to be visually similar. This similarity is expressed in an analytical way, which will afterwards allow us to use simple similarity metrics to reason about a recording's continuity.

In the case of video recordings, when discussing their continuity we can interpret L as the maximal pace of change for the analyzed type of video. Assuming that the discrete sequence (frames and their classifications) is a view of a continuous function (i.e. the real life view and state) satisfying the Lipschitz property, the appropriate limitations on the change over time are still kept. To show this, let us first define the function which represents the transformation of real-life (i.e. continuous) state into discrete video. A real life observation in a point of time t will be denoted as v_t. Furthermore, we will use m to denote a given frame number and assuming the first frame of the video starts at $t_0 = 0$, the time of frame m is

$$t_m = m \cdot step,$$

where *step* is the difference in time between consecutive frames. It is the inverse of the frame

rate, e.g. for a frame rate of 25 FPS, we have $step = \frac{1}{25}s$.

We will model the video recording with the function $V(\cdot)$ which satisfies:

$$d(V(v_t), v_t) < \frac{Q}{2}$$

for a given constant Q which expresses the inaccuracy of the transformation and reflects the quality of the recording. A discrete frame in a discrete moment m will be defined as

$$p_m = V(v_{m \cdot step}).$$

Figure 3.3 presents the transformation from real world continuous view into a discrete projection in both dimensions (time and observation). The figure presents two corresponding spaces:

(a) The observed reality is of continuous nature, both in terms of the time axis and the changes it undergoes.

(b) The video recording represents the reality as closely as possible. Still, frames are recorded only in regular intervals (*step* value on the time axis) and the acquired images are not fully accurate representations (observation axis - rounded values).

Therefore for two given observations v_s and v_t:

$$d(V(v_t), V(v_s)) \leq d(V(v_t), v_t) + d(v_t, v_s) + d(v_s, V(v_s)) \leq d(v_t, v_s) + Q. \tag{3.3}$$

This result corresponds to the fact that the difference between two frames represents the difference between the views they represent and a limited error of inaccuracy of the transformation (e.g. rounded colors, pixels etc). The higher the quality of the images, the lower the value of Q.

Theorem 2. A discrete view with limited inaccuracy of a Lipschitz continuous function preserves the Lipschitz property.

Proof. Let us define:

- v_t - observation in point of time t,
- $V(\cdot)$ - projection of a real life view into a picture/frame,
- $p_m = V(v_{m \cdot step})$ - discrete picture in the discrete moment $m \in \mathbb{N}$.

We want to show that the Lipschitz property is still preserved for p_m. Let us take arbitrary

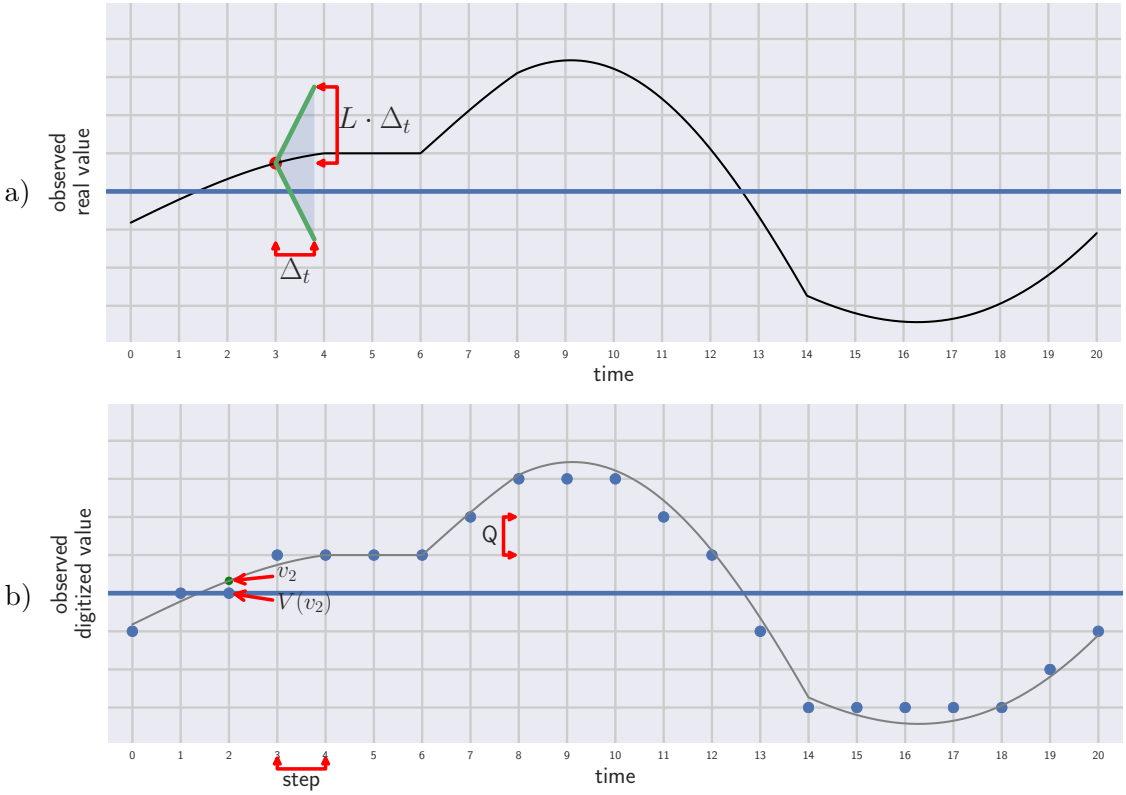


Figure 3.3 – The transformation of a real-life view into a video recording a) Observed reality b) Discrete and digitalized reality, where: step - time between two frames, Q - digitalization inaccuracy (i.e. pixels and discrete colors)

different frame indexes m and m'. Applying the previously defined properties we get:

$$\begin{aligned} d(p_m, p_{m'}) &= d(V(v_{m \cdot step}), V(v_{m' \cdot step})) \leq \\ &\leq d(v_{m \cdot step}, v_{m' \cdot step}) + Q \leq L \cdot step \cdot |m' - m| + Q \leq \\ &\textit{(Since } |m' - m| \geq 1\textit{, as they are different frame indices:)} \\ &\leq (L + \frac{Q}{step}) \cdot step \cdot |m' - m| = \\ &\overset{Q' = \frac{Q}{step}}{=} (L + Q') \cdot step \cdot |m' - m|. \end{aligned}$$

□

The proof leads us to another observation: To preserve the limit of change between consecutive frames the coefficient $r = (L + Q') \cdot step = L \cdot step + Q$ needs to be kept. What is most important to note is the fact that a faster pace of change can be compensated by a lower *step* value (i.e. higher frame rate). Therefore, the maximal pace of change for an observed view to be processed as continuous video is limited by current technical development (highest possible frame rate and image quality).

It is also important to note that the value of Q is far less influential, and is included in the calculations to account for minor digitalization noise. Since its value is representing visual distance between the perceived view and recorded images, Q is almost equal to 0 when an observer perceives the image as real-colored and cannot distinguish pixels.

3.1.4 Video stream and classification continuity conclusions

In the previous sections we have expressed the expected properties of classification functions as an analogue of the analytical continuity of the considered functions.

The second important issue that has to be faced is the difficulty of defining a proper d_Θ function - the "visual distance" metric in Θ. Constructing a "perfect" function that would meet both the definition of a metric and reflect the subjective visual similarity is virtually impossible. A number of works have already defined such functions for related usages [51, 52, 173, 174], such as segmentation of video sequences. Addressing this issue is one of the first steps for acquiring a method able to mimic the intuitive human understanding of similarity.

When nearby frames are compared in video sequences this problem becomes easier, as the differences and similarities are not of semantic nature but can be also expressed by simpler features (such as histograms). Going further, a very simple model based on the temporal distance between frames can already be expected to correlate with their perceived similarity.

It has to be pointed out again that Θ is in fact discrete and finite (as long as we can assume some limitation of the size of input images). It's only the human eye that perceives digital images of high quality almost as an exact representation of a continuous space of images. At the same time, the size of this space (i.e. number of possible input images) makes it computationally impossible to consider each of its elements individually (in a smaller space we could consider a predefined similarity value for all pairs of images). The number of possible True Color Full HD images amounts to an inconceivable amount[4]. Therefore, for the sake of establishing practical methods for measuring and evaluating particular classifying functions, applicable algorithms for computing similarity have to be defined.

Work with such simplified metrics has already been approached in [18], where simple histogram-based image similarity metrics have been used. The definition of the metric (difference of histograms) can be described as very forgiving, since it is robust to any kind of movement and slow color changes. Each subfigure of Figure 3.4 corresponds to one of three endoscopic films, whose frames have been classified using an algorithm [84] which returns a continuous value. The three plots show the dynamic of changes in each stage of the classification. The first represents the differences between consecutive frames, the second – the differences between frame signatures (feature vectors corresponding to the frames), the last one – the dynamic of changes of the classifications of frames.

When comparing the graphs, some correspondence in the dynamics of changes on all level can be observed or presumed, but not easily defined. The moments of rapid change in the graphs can

[4] at a 1920x1080 resolution with 24bit color: $2^{1920 \cdot 1080 \cdot 24}$

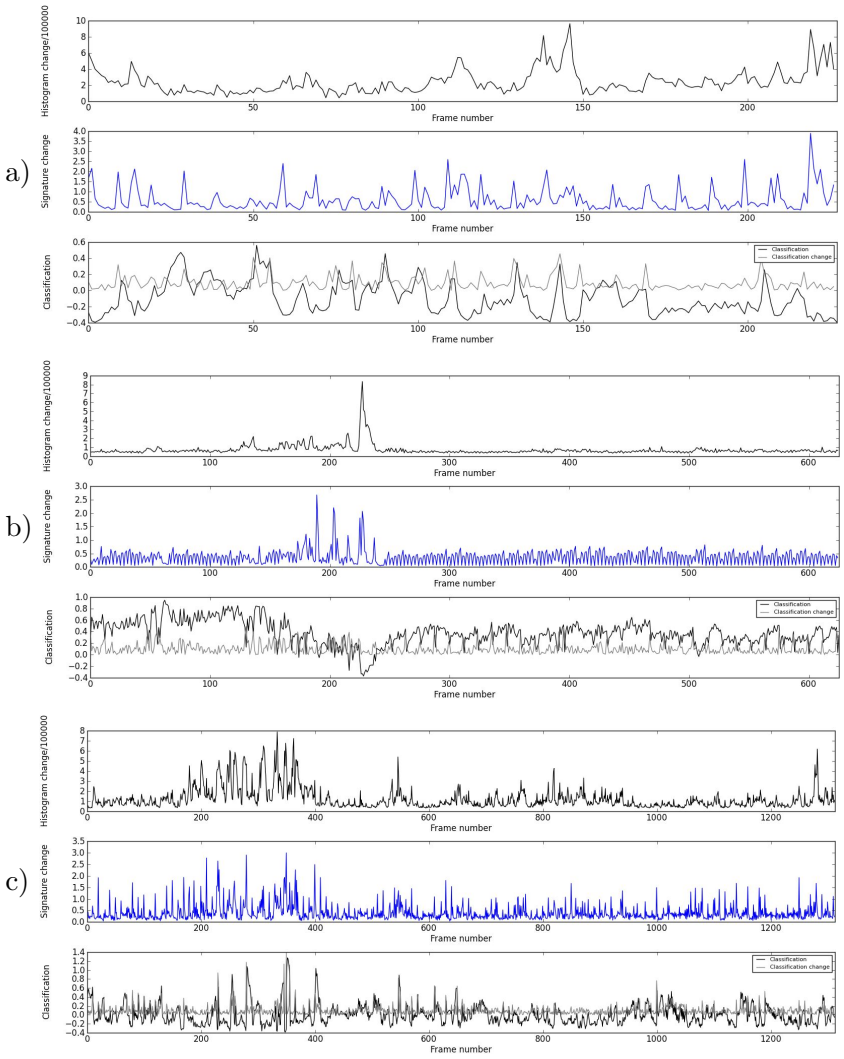


Figure 3.4 – Video/feature/classification change graphs for three exemplary videos. Source: [18].

be related to a number of factors (e.g. threshold values in the histogram definition, numerically unstable functions, overtrained classifiers, threshold values in decision functions). Each of them is a moment of disturbance in the corresponding sequence's continuity, and a point of particular interest for an FSA application. This interest comes from understanding that a single spike in a long sequence of equal values is more probable to be a mistake rather than a single outlier in the ground truth.

For our considerations we will further on use a metric function $d(\cdot, \cdot)$ which expresses the visual difference between its arguments, putting aside an exact derivation of its value unless specified otherwise.

This leads to a twofold understanding of video continuity, which depends on the context. First, we will call a video **continuous**, if the function $d(\cdot, \cdot)$ is defined and for the consecutive

frames of the video stream $\{p_m\}_{m\in\mathbb{N}_0}$ we have:

$$\forall_{m\in\mathbb{N}_0} d(p_m, p_{m+1}) \leq L \tag{3.4}$$

for a value of L which is lower than the maximal possible difference between frames. This ensures that the criterium does not accept all videos.

For real-life videos, which can be perceived as continuous, we assume that an implicit similarity function fulfills this condition. In the case of the test data, which has been used in the experiments performed within this work, the evaluation of exemplary similarity measures in Figure 5.14 confirms the g-continuity of the evaluated datasets.

3.2 Preliminary experiment

This section contains the results of [19], provided without any alterations. It illustrates another concept of applying temporal relations in video than that approached in this dissertation. Its results allow to observe the level of relevance of frames within small neighborhoods. Furthermore, some techniques from this experiment have been also used as building blocks of the fFSA method proposed in the next chapter.

Until this point we have established reasons to analyze results of OFA classifications taking into consideration their temporal structure. The experiment from the paper [19] proposed an approach which was build on similar premises with the goal of accelerating the classification of frames in video sequences.

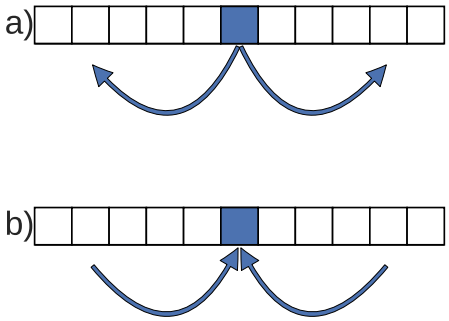


Figure 3.5 – Different approaches of applying temporal relationships.

Two concepts of considering the relation of a window and its central frame have been presented in Figure 3.5. The variant a) has been applied in the experiment in the paper - it involves a variable-sized window, which is classified as a whole using the central frame's classification. The variant b) where a single frame's classification is influenced by its surrounding window will be the one which we will further analyze in this work.

Nevertheless, in this section we will take a brief look at the experiments from [19], to get a better perspective on the tradeoffs and benefits of the two approaches.

Let us define the following functions for comparing frame similarity and therefore determining each window's size:

- Simple distance (SD) - the L^1 distance between the images (vectors of pixel values),
- Simple distance on processed image (SP) - the Simple distance of two images after blurring and downscaling,
- Histogram distance with k bins (HD, HD_k) - the L^1 distance between concatenated k-bin histograms of all channels of the images.

For an easier comparison and evaluation, the metric values are linearly normalized to the range $[0; 1]$ (where 1 means the maximal possible value of the metric for the considered image size).

In the proposed algorithm consecutive windows are established by repeatedly executing the following two steps:

- **Expansion** - consecutive frames are assigned to the scene until a frame is reached whose difference from the specific frame exceeds a given threshold. That frame will be the next specific frame.
- **Reduction** - frames from the end of the current scene, which are more similar to the next specific frame than to the current one are reassigned to the next scene.

The pseudocode (exact with respect to treating some boundary cases) of the algorithm has been presented in Algorithm 1.

Experiments

The performed experiments involved the evaluation of the algorithm on a set of exemplary recordings from conventional endoscopic examinations. The videos have been acquired at 25FPS, with a resolution of 720x576 pixels. The classified binary property annotated in the ground truth was the presence of lesions (true) or only healthy tissue (false) in the view. Six representative films have been chosen, fulfilling the following criteria:

- at least 1000 frames long,
- the recognized property is present in 20% to 80% of the frames,
- there are at least 5 changes of the classification in the ground truth.

Those requirements have been set to prevent the overrating of algorithms which would just propagate a single result on all frames. The videos used in the experiment had altogether 7273 frames.

The scene segmentation algorithm has been applied to every film, using five different metrics with thresholds adjusted to their characteristics (SD and SP are showing larger changes in

Algorithm 1 Scene segmentation

Input:
F - sequence of N frames
T - threshold value for metric d
Output: S - scene assignments
$scene \leftarrow 1, current \leftarrow 1, next \leftarrow 1$
while $current < N$ **do**
  #Expansion
  while $d(F[current], F[next]) < T$ and $next < N$ **do**
    $S[next] = scene$
    $next \leftarrow next + 1$
  end while
  $scene \leftarrow scene + 1$
  $mid \leftarrow next$
  #Reduction
  while $d(F[mid], F[next])$
      $< d(F[mid], F[current])$ **do**
    $S[mid] = scene$
    $mid \leftarrow mid - 1$
  end while
  $current \leftarrow next$
end while

value). Then, the ground truth classification values of the specific frames have been assigned to their corresponding scenes. This imitated a perfect underlying classifier, which may be to computationally expensive to call for every single frame.

Results

The first observation of the evaluation is the relatively high distance in SD and SP metrics for seemingly similar images. These metrics turned out to be very sensitive to even minor shifts and to be applicable mostly for images with a very high amount of common static areas.

For various tested threshold values the average scene lengths have been computed and the results presented in Table 3.2 and Table 3.3. As expected, a clear positive correlation between the threshold value and scene length can be seen for all metrics.

Table 3.2 – Average scene lengths for HD_k metrics.

Threshold	0.05	0.1	0.15	0.2	0.25	0.3
HD_4	3.2	6.2	10.0	15.3	21.9	29.8
HD_6	2.8	5.0	8.1	11.8	16.7	23.1
HD_8	2.5	4.5	7.2	10.1	14.2	19.9

Figure 3.6 presents the relation of recognition accuracy and metric thresholds. It is important

Table 3.3 – Average scene lengths for SD and SP metrics.

Threshold	0.4	0.42	0.44	0.46	0.48
SD	32.6	73.3	91.3	131.8	177.8
SP	16.3	23.1	32.1	48.2	57.7

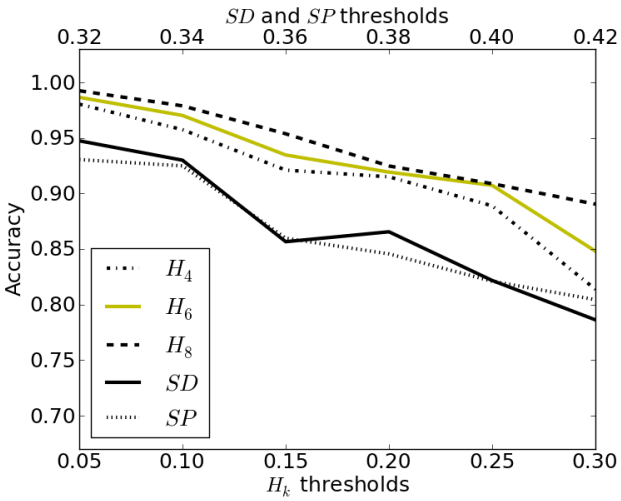


Figure 3.6 – Accuracy change with threshold.

to note, that only the values of the HD_k metrics are directly comparable in this graph, due to the diverse definitions of the metrics.

The graph in Figure 3.7 presents the relation between the acquired average scene sizes and the corresponding classification accuracy. It can be seen that all of the H_k metrics acquired similar results and outperformed the SD and SP metrics. This shows that the scene segmentation algorithm with the H_k metrics acquires a better division into scenes and assignment of their specific frames.

High accuracy values of over 95% are preserved for scene sizes of up to six frames. With such results, costly recognition algorithms might be improved to operate on whole scenes. The scene segmentation algorithm can be tuned in respect to the given time limitations depending on an accepted performance/accuracy trade-off.

Summary

In the paper [19] a new method for accelerating the classification of frames in video sequences has been proposed, which is based on the same observations as the motivations for this work.

The accelerating method is trading result accuracy for algorithm efficiency (assuming that a computationally expensive OFA algorithm was used). Further on we are going to propose the new FSA approach, which will aim to improve classification quality without impeding the whole classification's efficiency.

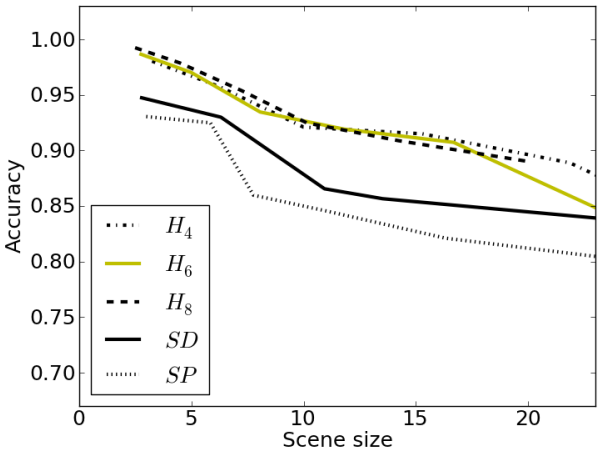


Figure 3.7 – Accuracy change with scene size.

3.3 Probability of decision rule correctness - discussion

In this section we will analyze how the probabilities of acquiring a correct result depend on whether a single frame or its whole neighborhood is taken into consideration.

The OFA accurracy measures (true/false positive/negative) are assumed to be known beforehand. We will further label them as follows (using 1 for positive and 0 for negative):

		Actual Value	
		Positive	Negative
Classification	Positive	$R_{1/1}$	$R_{1/0}$
	Negative	$R_{0/1}$	$R_{0/0}$

First we will consider a window of width $w = 2k + 1$. The scene length is a random variable $\mathcal{Z}$ with an unknown distribution. This distribution is dependent on the given problem and characteristics of its data. Although, due to the continuity of the video, we can assume that its average value is significantly larger than 0. We choose such a maximal window size $w_{\max}$ that ensures that the vast majority of windows are fully contained in a single scene:

$$P\left(w_{max} \ll \mathcal{Z}\right) \approx 1. \tag{3.5}$$

The OFA methods considered for the FSA schemes are expected to be methods of relatively high accuracy. Because the rate of positive and negative frames in the whole sequence is unknown, we need to assure that both $R_{0/1} \ll 1$ and $R_{1/0} \ll 1$. Therefore, in further considerations we may assume that $R_{0/1} \approx R_{1/0} \approx R_F = \max(R_{0/1}, R_{1/0})$. Such a simplification is fully justified, as the distribution of positive/negative inputs in the ground truth is unknown and in the worst case could be dominated by the one of the two values, which is more difficult to classify.

Therefore, the probability of a single value being correct is

$$P(O_m = G_m) = R_{G_m/G_m},$$

where:

- O_m - OFA classification at discrete point in time m (algorithm's answer),
- G_m - ground truth at point m.

The probability of all values in a given window range being correct is:

$$P(\forall_{i\in[m-k\ldots m+k]} O_i = G_i) = \prod_{i=m-k}^{m+k} R_{G_i/G_i} \approx (1 - R_F)^w. \tag{3.6}$$

The probability of acquiring a given number s of ones (positive classifications) in a window of width $w = 2k + 1$ comes from the observation that for both possible versions of the underlying ground truth ($G_m = 0$ or $G_m = 1$) the number of ones has a binomial distribution:

$$\begin{aligned} P(\sum_{i=m-k}^{m+k} O_i = s) &= \sum_{c\in\{0,1\}} P(\sum_{i=m-k}^{m+k} O_i = s | G_m = c) P(G_m = c) = \\ &= \sum_{c\in\{0,1\}} \binom{s}{w} R^s_{1/c} \cdot R^{w-s}_{0/c} \cdot P(G_m = c). \end{aligned} \tag{3.7}$$

Assuming that Equation 3.5 is fulfilled, we get that the currently observed window:

- is fully contained in a single scene - in virtually all cases,
- contains a scene boundary, i.e. its frames belong to two consecutive scenes.

Any other situations would amount to a negligible number of cases (as the maximal window size has been chosen in accordance with Equation 3.5) or a non-continuous segment of the video.

Continuing, we will establish the confidence in the decision indicated by a majority vote in a shifting window. It can be defined as the probability of the underlying ground truth being equal to c (0 or 1) given that the window contains s positive classifications (using Bayes' theorem):

$$\begin{aligned} P(G_m = c \mid \sum_{i=m-k}^{m+k} O_i = s) &= \frac{P(\sum_{i=m-k}^{m+k} O_i = s \mid G_m = c) P(G_m = c)}{P(\sum_{i=m-k}^{m+k} O_i = s)} \\ &= \frac{P(\sum_{i=m-k}^{m+k} O_i = s \mid G_m = c) P(G_m = c)}{\sum_{c\in\{0,1\}} P(\sum_{i=m-k}^{m+k} O_i = s \mid G_m = c) P(G_m = c)}. \end{aligned} \tag{3.8}$$

Exemplary results of Equation 3.8 and Equation 3.7 have been presented in Figure 3.8. The numerical values provide strong indication for leaning towards the majority result when deciding on the window center's classification assignment. Such an approach can be described as the **simple voting approach** because the majority vote is performed in a straightforward manner,

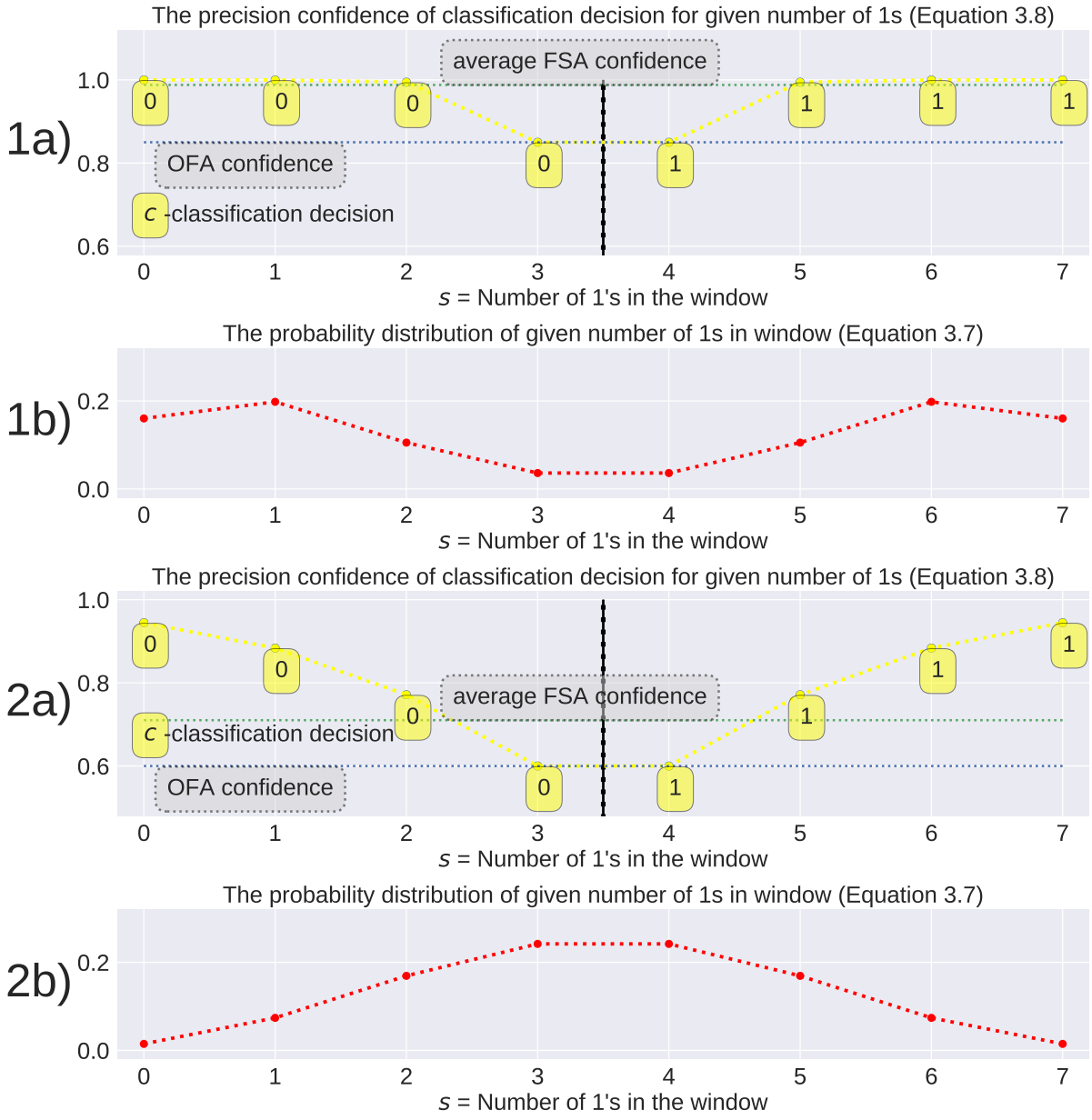


Figure 3.8 – Probabilities of True Positive or True Negative findings depending on the number of frames matching the considered classification. Window size is $w = 7$; ratio of positive values in the ground truth $= 0.5$, ratio of negative values $= 0.5$; accuracy parameters: Case 1: $R_{1/1} = R_{0/0} = 0.9$; Case 2: $R_{1/1} = R_{0/0} = 0.6$

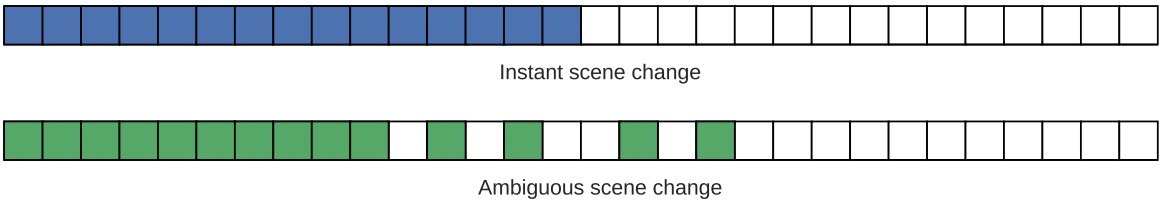


Figure 3.9 – Scene change types.

without taking additional information into consideration. Possible extensions will be discussed in the next section. The confidence of such a decision is at least at the level of OFA classifications.

The results presented in the figures correspond to the vast majority of cases, when the shifting windows is fully contained in a single scene (in those cases for which $\forall_{i\in[m-k\ldots m+k]}G_i = G_m$, $G_i \in \{0, 1\}$). Our assumption about the relation of window and scene sizes implies that in the continuous segments of video windows with single scene changes can happen, although rarely. This can proceed in two ways (as presented in Figure 3.9):

- instant scene change - an immediate change from a sequence of positives (negatives) to a sequence of negatives (positives),
- ambiguous scene change - an interim phase with low confidence levels and multiple changes between positive and negative ground truth classification values.

In the first case, small discrepancies between the classifications and ground truth can be reason enough to mislead the FSA reasoning into introducing incorrect changes. Still, the introduced mistakes would result only in minimally moving the border between two scenes.

In the second case, even perfect classification results can be perceived very bad ones and it is more difficult to apply FSA reasoning. This corresponds to the situation presented in the second example in Figure 3.9. Such an ambiguous transition between scenes can be encountered in multiple real life datasets due to multiple reasons:

- Objects remaining at the edge of visibility - e.g. a person standing at the edge of the camera's view.
- Ambiguities of classification criteria - even when annotating a dataset it might be a matter of opinion e.g. at which point a person entering the view is already considered visible (shadow? arm/leg? face? whole silhouette?).
- Unstable behavior of the classifying algorithm - this case is related to the previous one, but considers the output of algorithms which might behave unpredictably in case of partially visible objects.

Extending the simple voting approach

The considerations above concentrated on a simple vote performed with the classifications of frames in a shifting window. This way the frames just next to the frame of interest have the

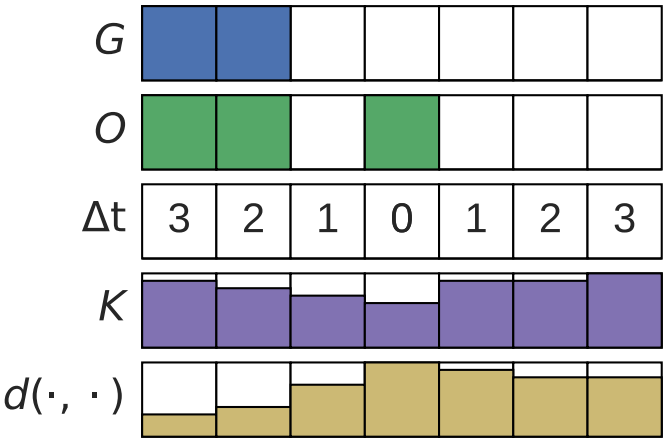


Figure 3.10 – The information provided for reasoning in a shifting window.

same influence on the vote outcome as those at the ends of the window. The fact that the view in the frames changes gradually has not yet been taken into consideration.

Furthermore, possible OFA algorithms can also provide additional information regarding the confidence of a given classification. That way a positive or negative with 100% certainty can be treated as a stronger indication than one with 55%, which can be reflected in the weights of the vote.

In the case of taking a more extended approach, we can also consider the classified frames themselves as a source of information for the FSA scheme. On the one side the similarity of a given frame to the central one can indicate that its classification is also more probable to be similar. Still, on the other side, a large similarity distance between frames in a temporal proximity can indicate a rapid change of the view, a point of discontinuity in the video or the presence of a sudden distortion in one of the frames.

Figure 3.10 presents the different kinds of information which can be provided for frames in a shifting window. Those are:

- G - the ground truth (available in the training phase),
- O - the classifications acquired from the underlying OFA algorithm,
- Δt - distance from current frame (in number of frames or seconds),
- K - own certainties of choice (available if provided by underlying OFA algorithm),
- $d(\cdot, \cdot)$ - similarity to central frame (computed and used only in the fFSA approach).

It is important to note that all FSA algorithm classes which utilize the OFA classification sequence O as well as any additional information also contain the simple voting scheme as their representative. This is ensured by disregarding any data other than O and performing a simple vote regardless of the additional information. Therefore the discussion in this section will hold when more elaborate classes of algorithms are considered, with the performance of the simple voting method being a reasonable lower estimate of their performance.

4 The FSA approach

In this chapter we start with describing the FSA approach in a greater detail, explaining the two variants of the FSA method - iFSA and fFSA. Next, the general outline of FSA methods is presented, together with a discussion of the computational complexity it adds to the classification task. Afterwards component functions for FSA algorithms are discussed and multiple possible variants are noted - out of which a number will later be selected for implementation and experiments. In the end of the chapter evaluation criteria are defined for comparing algorithms classifying images in video streams with each other. Those focus both on the accuracy of frame-by-frame classification (FPR, FNR- false positive/negative ratios) and on the stability of the resulting temporal binary sequence (IBR, MBR- invalid/missing boundary ratios).

4.1 Main methods

In the previous chapters we have analyzed works related to classifying frames and images in videos and approached the subject of classification improvement using a shifting time window approach. We can therefore define methods based on that approach, as presented in Figure 4.1.

One Frame Analysis (OFA)

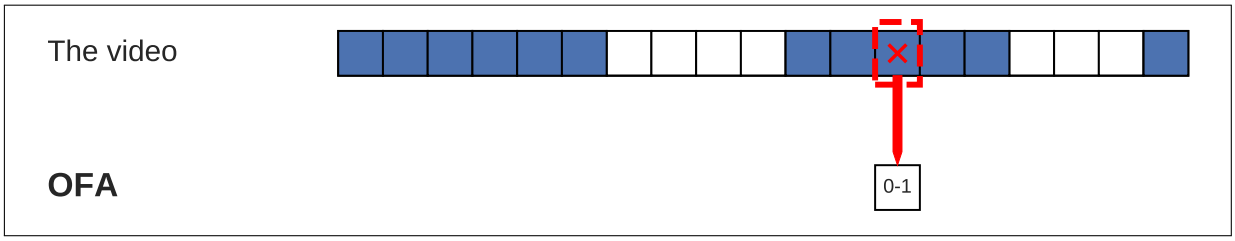
Input: *A single frame*
Output: *Classification of the frame*
The OFA method is the base of the considered field. There are numerous methods which classify static features, used both in video analysis as well as straightforward image classification. The OFA method has been presented in Figure 4.1a.

Furthermore, OFA is used as a component of both FSA variants: iFSA and fFSA, therefore its classifications can be found as input data in their corresponding illustrations.

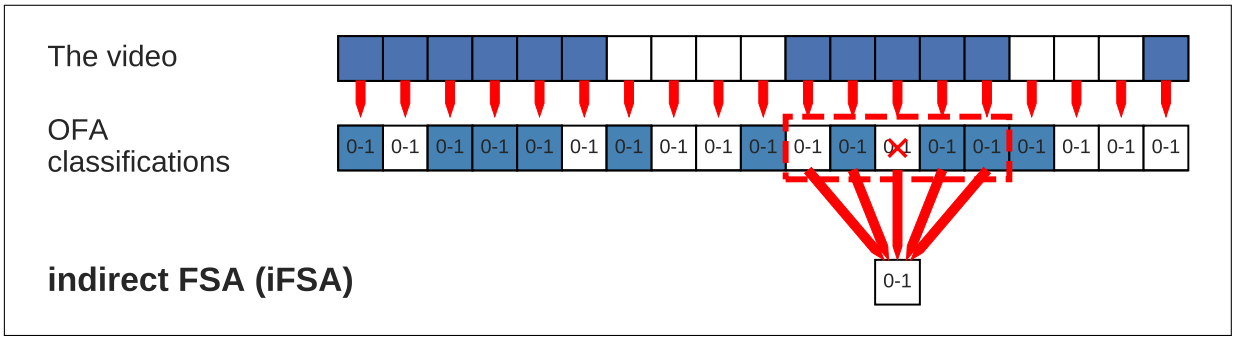
Indirect Frame Sequence Analysis (iFSA)

Input: *Classifications of a segment of a video centered on the classified frame*
Output: *Classification of the central frame*
Indirect Frame Sequence Analysis is presented in Figure 4.1b. Simple versions of it are informally used in a number of works, although usually only on the basis of intuition.

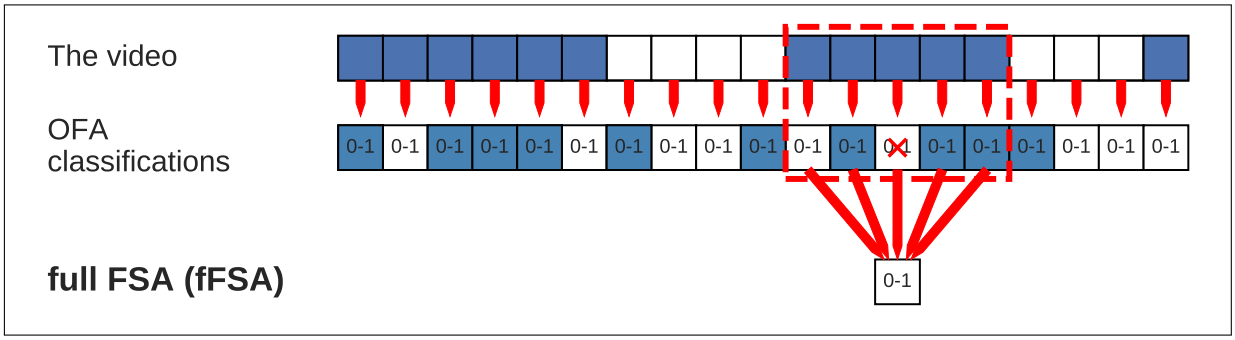
The method consists of two steps: preliminary OFA classification and the FSA step (processing of the OFA output). The input of the latter is only a time sequence of classifications. Therefore



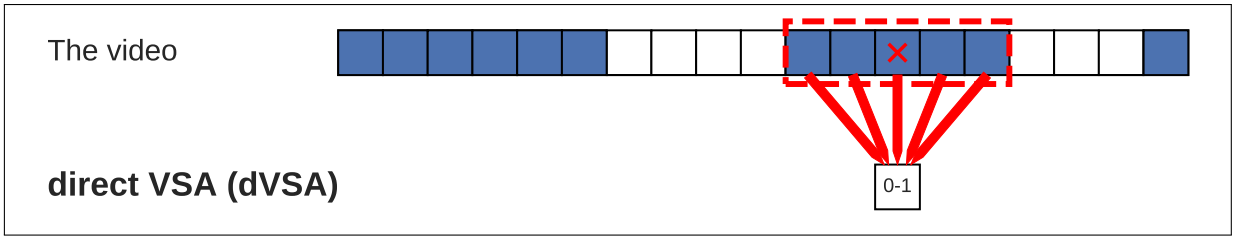
(a) The OFA approach: one frame in the input video is classified independently of all others.



(b) The indirect FSA approach. Each frame of the input video gets a preliminary classification from an OFA method. The classifications in the frame's neighborhood influence its classification.



(c) The full FSA approach. Each frame of the input video gets a preliminary classification from an OFA method. *Both* the classifications and frames in the frame's neighborhood influence its classification.



(d) The direct video segment analysis approach. The whole video segment centered on a frame is directly analyzed to provide the frame's classification.

Figure 4.1 – A comparison of the discussed classification approaches. The two FSA variants are an original solution of this dissertation. The dVSA is noted for the discussion of possible approaches.

the FSA step is abstracting fully from the video provided to the underlying OFA classifier.

Full Frame Sequence Analysis (fFSA)

Input: *A segment of a video centered on the classified frame and its sequence of classifications*
Output: *Classification of the central frame*
The second variant of the FSA approach, Full Frame Sequence Analysis (Figure 4.1c) incorporates a segment of the input video into its FSA step's reasoning. Therefore, additional information about the video can be taken into consideration. For instance, the visual similarity between frames in the time window can allow to predict which classifications are more relevant to the one which is in the center.

As such, the fFSA method can be considered a superset of iFSA, since every iFSA method can be presented as a fFSA method disregarding a part of its provided information.

Direct Video Segment Analysis (dVSA)

Input: *A segment of a video centered on the classified frame*
Output: *Classification of the central frame*
The dVSA method, presented in Figure 4.1d, involves creating a vector from a whole considered window of images/video frames (or their features) and classifying it by a generalization of its corresponding OFA algorithm.

In the case of the dVSA method one has to consider the issue of the high data dimensionality, which is known to introduce additional problems to classification tasks (e.g. the *curse of dimensionality* phenomena [10, 149]). Because of this and due to the costly training data preparations (need to label a representative number of diverse video segments) it is a more hypothetical proposition, which is noted but won't be a subject of this work.

OFA methods are a broad area of research and they are being successfully used in numerous fields. Training classification algorithms based on single images is cheap due to the availability of free sources of data and low cost of manually creating a base with an appropriate number of cases (e.g.[50]). Mostly, OFA methods are developed for specific domains (e.g. using features which relate to the domain).

Numerous works listed in subsection 2.2.1 apply methods corresponding to the FSA approach – they utilize the results of OFA algorithms, which are considered as a temporal sequence for further improvement. Most of the cited methods can be assigned to the *indirect FSA* category. Their application is based on simple intuitions, without elaborating on their influence on the results or correctness. Therefore, there is still a lack of a unified definition and analysis which would constitute a solid reference point for applying such methods.

So far, no methods which could be classified as fFSA have been proposed. Because such an idea allows to combine the advantages of cheap OFA training and broad reasoning based on the

temporal relations between frames, it is reasonable to evaluate the improvements in classification quality of such a class of methods.

4.2 General FSA algorithms' outline

The main goal of the the FSA approach is to provide an efficient and effective scheme for improving an incorporated OFA algorithm. The task-specific classifier training is performed on a single-image level and the post-processing FSA step uses reasoning based on the properties of a continuous video stream.

Such an approach turns out to be especially valuable if we consider the alternative option of using spatio-temporal methods trained on whole video sequences or parts of them. This alternative would introduce an additional cost of preparing a training set consisting of a significant number of fully classified videos - either on a per-frame basis (providing classifications for all frames in a video) or even with full spatio-temporal boundaries (annotating the exact properties, e.g. position, of the sought features in classified frames). To avoid the cost of preparing an appropriately large dataset, an underlying OFA classification will remain to be the core 'classifying' step of every FSA algorithm resulting from the FSA approach. As it was already stressed in Chapter 2, OFA methods provide means for cheaper training and plenty of annotated datasets from multiple domains.

What is more, the OFA algorithm is actually treated as a black-box providing classifications for frames, but without any knowledge about its method being exposed. Therefore in the end, the underlying OFA methods for a given problem can be interchanged and the resulting FSA algorithms compared with each other as long as they are evaluated on the same input.

The most general outline of an FSA algorithm is as follows:

FSA algorithm outline

1. **Input:** Video sequence F.
2. All frames in F are classified using the incorporated OFA algorithm.
3. **Intermediate result:** Sequence of preliminary classifications.
4. The preliminary classifications sequence is adjusted with regard to the proximity and similarity of nearby frames.
5. **Output:** Sequence of final classifications.

Step number 4 is where the chosen FSA scheme is applied, transforming the preliminary classifications into final ones. Its pseudocode has been presented in Algorithm 2. The composition of the OFA classifier and the FSA scheme together forms the final **FSA algorithm**. Interchanging the FSA schemas for the same OFA algorithm allows to compare their improvement rate with each other. On the other hand, when a schema is tested on multiple OFA algorithms,

Algorithm 2 FSA step pseudocode

FSA step:
Input: Sequence of n OFA classifications O (possibly: with accompanying frames)
Output: Sequence of n FSA classifications C
Parameters:
- w - window size,
- A - acceptance threshold,
- decision function specific parameters

Algorithm:

1. **Init** C with O
2. **For** every window of width w in O:
3. Perform a parameterized **decision** on the classification of the central frame
4. **If** the decision result exceeds the acceptance threshold A
5. **Change** the current central frame's classification in C
6. **return** C

its applicability to particular kinds of problems can be evaluated.

The classification of a frame is adjusted according to its closest frames in the sequence. We defined the **shifting time window** as a sequence of consecutive frames centered on the the current frame of interest. The function measuring the similarity between two images will further be called a **similarity metric** - it has to be taken into consideration though, that the function actually might not fulfill the mathematical definition of a metric, as it will be discussed further on. We will denote it as $d(\cdot, \cdot)$ and may put aside its exact derivation.

In some cases the actual outline of an FSA algorithm might only logically comply to that presented above, mostly for performance reasons. Still, in all cases it is possible to implement a less time-efficient version suiting the general FSA outline.

4.3 Components of an FSA algorithm

The general description of FSA methods needs to be complemented with information about the considered controlling components of the algorithms. Their complete list can be found below:

- built-in OFA algorithm (+ its parameter set, all in one black-box);
- time window parameters:
  - window size (number of frames),
  - window symmetry (i.e. the number of frames before or after the considered one),
  - rules defining the time window size:
    - fixed size,
    - window size and symmetry varying (adapting), considering:
      - local frame similarity/pace of change in video,
      - desired delay (adjusts right side of window),

  - dynamic throughput increase - adapting window size to proceed in real-time;
- FSA decision rule:
  - vote threshold,
  - weight distribution based on the frame distance/similarity,
  - weight distribution based on the preliminary classification's quality/confidence,
  - dynamics considered (first/second derivatives, difference quotients etc.),
  - regression/approximation,
  - probabilistic model;
- image similarity metric used (for fFSA).

This extensive list contains all components required to fully define an FSA algorithm - both those covered in this dissertation as well as other ones. For many algorithms some of those options are irrelevant or disregarded (a trivial default choice is kept), resulting in that there might be no necessity to explicitly set all options. At the same time the exchangeable components, and broad ranges of possible OFA algorithms and metrics ensure the necessary flexibility of the presented class of algorithms for handling a wide range of different problems.

It also has to be pointed out that for some cases there is more than one way of describing an algorithm (e.g. smaller window or zero weights on the edges of a bigger one). Further on such ambiguities will be avoided, by choosing a single representation of an algorithm whenever possible. Minimalistic descriptions will be preferred, limiting the size of the shifting window and amount of superfluous parameters (e.g. a single multiplier on all weights).

In the subsections below each of the considered FSA components with their related parameters and variables is discussed in depth.

4.3.1 Built-in OFA algorithm - O, K

Each FSA algorithm is acquired by applying the FSA step which transforms the sequence of the OFA algorithm's output classifications O into the final classifications C. The underlying OFA algorithm, besides providing just the sequence O, can also return the classification confidence values K, where $K_m \in [0; 1]$. K_m should be interpreted as an analogue of classification probability correctness for frame m, i.e. $K_m \approx 1$ indicates a maximal certainty of the classification.

Multiple algorithms can be classified as representatives of the OFA algorithm class. They are well researched and constantly improved with new approaches [43, 104, 121, 132, 136, 142], coming from many domains, such as e.g. medical image recognition, CCTV systems or gaming.

The accuracy of the used OFA algorithm is the baseline for evaluating the improvement introduced by the FSA method. The chosen evaluation measures show how many mistakes each algorithm makes. Rather than evaluating a single FSA algorithm's performance, we are going to focus on the ratio of errors remaining after the FSA scheme has been applied.

As it was stated earlier, an OFA algorithm classifies each frame separately and an instance of the algorithm remains stateless (that is - there is no information stored between classifications).

As the FSA method cannot improve a sequence of classifications that is pure noise, it should be ensured that the accuracy of the underlying OFA method is high enough. As discussed in Chapter 3, with a perfectly uniform distribution of classification errors, an accuracy of over 50% would be enough to ensure that the FSA approach would improve the classification results. Since sequences of higher and lower accuracy can appear, higher values of general accuracy of the OFA algorithm need to be ensured so that the overall result is also improved. Therefore the FSA methods will be evaluated on OFA algorithms which acquired an accuracy of over 70% in published works.

Besides these two requirements (statelessness and a sufficiently high accuracy), we assume a black-box approach regarding the details of the OFA algorithms.

4.3.2 The time window - w, $w_{\max}$

The time window is the neighborhood of the current frame, which is considered for possibly altering that frame's classification. The simplest windows are those found in the aforementioned works – symmetrical windows of a very small size, used to even out minor misclassifications.

In the more general case, we can specify a number of parameters which can define a window. A single window can be simply described as the number of frames before and after the currently considered one. In a video sequence the consecutive windows can be defined with those parameters either fixed or varying depending on factors such as:

- the pace of change in the frame's neighborhood - the window can be expanded if the current scene has a small pace of change,
- the confidence of classifications in the window - if the classification confidence of the OFA algorithm is available, the window size can be adjusted according to the confidence distribution, e.g. to leave out ambiguous sequences,
- adapting for throughput and delay requirements - when processing a video stream in real time, the width of the window on the right side directly influences the minimal delay, while the full window size might influence the processing throughput.

4.3.3 Image similarity metric - $d(\cdot, \cdot)$

The image similarity metric is a tool to be incorporated into some of the decision rules for the fFSA variant of the FSA method. It is listed as a separate parameter to allow for the interchangeability of the various alternatives for any given decision rule.

We have intuitively defined a similarity metric as a function of two images which returns a non-negative value interpreted as the visual distance between those images. Below follows the full analytical definition of a metric function.

Definition 4. We call a given function $d : \Theta^2 \to \mathbb{R}$ a **metric** if for all $a, b \in \Theta$ it satisfies the following conditions:

1. $d(a, b) \geq 0$
2. $d(a, b) = 0 \iff a = b$
3. $d(a, b) = d(b, a)$
4. $d(a, b) \leq d(a, c) + d(c, b)$

As it was stated before, the functions we use are not necessarily required to strictly fulfill all of the conditions above. Most importantly, throughout this dissertation we will use a more loose formulation of the second condition in the form:

2'. $d(a, a) = 0$

possibly allowing for the function to represent two different images as not different at all. If all other conditions are fulfilled, such a function is called a **pseudometric**. Even though a pseudometric does not allow to establish the identity of an image, it can provide the necessary expression of similarity distances between given images.

Because of the subjectivity of visual similarity, possible computing rounding errors and the open formulation of new metric functions, in practice we might also allow for the metric function to occasionally violate the triangle inequality. A function fulfilling only the conditions 1-3 from the definition above is called a **semimetric**. Those conditions themselves describe just a function assigning positive numbers to pairs of values, without any relation between them. This is why semimetrics of our interest would still need to be restricted by a version of the triangle inequality, albeit weaker. Such a restriction could be realized e.g. as a tolerance level, accounting for the possible numerical effects causing a given metric function candidate to break the triangle inequality. For the experiments performed for this work, such functions have been avoided, therefore no deeper analysis of this subject will be provided.

In Figure 4.2 a number of simple examples illustrating those types of metrics have been shown. They demonstrate how common operations like rounding or thresholding affect the properties of functions, making them break some conditions of the metric definition.

Continuing, we will now discuss the desired properties of a similarity metric for practical applications. Perfectly, it should be insensitive to minor differences such as:

- pixel changes,
- small shifts and rotations,
- transformations preserving the view, e.g. brightness/contrast changes, small blur, etc. . . ,

and consider negligible alterations of a single image to be very close to each other. Such a similarity could be established by the proximity of the feature vectors, which correspond to an abstract description of the compared images. As it has been shown in [18], during tests on real-life videos well established feature vector functions don't always perform in the desired way, but tend to change value rapidly on neighboring frames as well.

- Pseudometric
  - $d((x_1, y_1), (x_2, y_2)) = |x_1 - x_2|$ - one of the coordinates is not important for the metric
  - $d(x, y) = |\lfloor \frac{x}{10} \rfloor - \lfloor \frac{y}{10} \rfloor|$ - the digit in the units position does not matter for the metric, e.g.:
$$d(12, 18) = |1 - 1| = 0$$
- Semimetric
  - e.g. a function assigning numerical labels depending on the arguments similarity (0-same, 1-close, 10-different)
$$d(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } |x - y| < 10 \\ 10 & \text{else} \end{cases}$$
$$d(1, 15) = 10 > 1 + 1 = d(1, 10) + d(10, 15)$$
- Premetric (both pseudo- and semi- at once)
  - $d(a, b)$- number of decimal positions, where a and b differ by more than 1.
$$d(111, 333) = 3 > 0 = 0 + 0 = d(111, 222) + d(222, 333)$$

Figure 4.2 – Examples of pseudo- and semimetrics.

Another approach allowing to disregard minor changes between compared images is to operate on downsampled and downscaled versions of the original images, which are then compared as vectors. Since both those operations involve rounding of values, care has to be taken to prevent actually increasing the perceived amount of change on threshold values.

The incorporation of the similarity metrics in fFSA methods allows for putting more focus on basic similarity functions, without having to look for such which could be directly applicable for the classification. Since the classification has already been performed by the underlying OFA algorithm and it can be assumed on the ground of the video's continuity that nearby frames have virtually the same features, the metric used in the fFSA reasoning has only a significance for adjusting the weights in the shifting window.

Possible approaches for computing the similarity of two given images include:

- a simple difference (in an L^p norm) of the images,
- the *Mahalanobis distance* [9], which can be defined in the algebraic form:
$$d_{\text{Mahalanobis}}(\theta_a, \theta_b) = \sqrt{(\theta_a - \theta_b)^T \mathbb{M} (\theta_a - \theta_b)}$$
with a symmetric positive semi-definite matrix $\mathbb{M}$, which corresponds to applying a linear transformation $\mathbb{S}$ ($\mathbb{M} = \mathbb{S}^T\mathbb{S}$) to both vectors. Such metrics can either be explicitly defined or obtained from training data, by the means of metric learning [28, 165, 170].

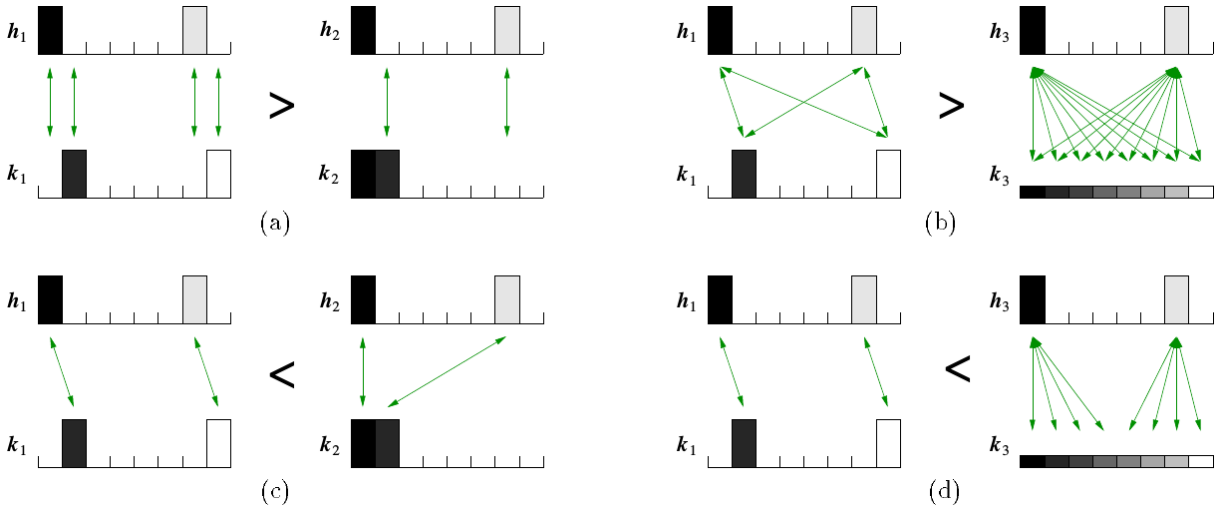


Figure 4.3 – EMD - perceived difference. a) L^1 b) Mahalanobis. c) and d) - corresponding EMD evaluations. Source: [137].

- Besides methods applying linear transforms, also non-linear functions may be applied if we present the above metric in a more general form with a given function f:

$$d_m(\theta_a, \theta_b) = \|f(\theta_a) - f(\theta_b)\|_2 \tag{4.1}$$

  This formulation brings this approach very close to the previously mentioned preprocessing of images before comparing them. The possible pre-processing functions f can be:

  – color histograms,
  – texton histograms,
  – self-correlation,
  – LBP textural features [122] extracted at multiple resolutions and in different color spaces [116],
  – localized LBP-based statistics [62],
  – statistics of cooccurrence matrices [113],
  – Gabor features [134].

- the Earth Mover's Distance (EMD, [137]) between histograms and histogram-grids. The EMD allows for a more fine grained concept of distance for comparing distributions. This is acquired by applying the idea of perceived similarity. The concept has been presented in Figure 4.3. The L^1 measure used in example a) and the Mahalanobis distance in example b) evaluate the pair of relatively similar histograms (left side in both cases) as more different than the given examples. When the EMD distance is used, the more similar pairs of histograms are properly identified, as it is shown in the corresponding examples c) and d) below.
  The perceived distance of different coordinates in the vector is provided as a parameter

of the method. Therefore, the EMD can be computed for histograms in color spaces with polar coordinates, like HSV (Hue-Saturation-Value) which resembles the human perception of images.

The general EMD method is computationally expensive (between $O(N^3)$ and $O(N^4)$ for comparing histograms with N bins) and is therefore limited in its application. Algorithm for particular bin-to-bin distances can be developed, which exploit the properties of those distances [98];

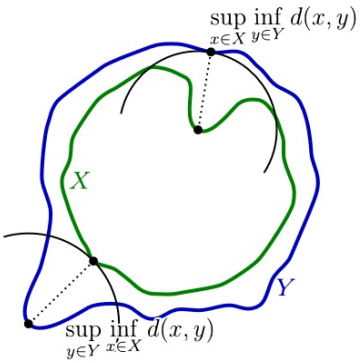


Figure 4.4 – Hausdorff metric evaluation example [a]

[a] Licence: Rocchini/CC-BY3.0 Source: https://commons.wikimedia.org/wiki/File:Hausdorff_distance_sample.svg

- Edge distance based methods - e.g. Chamfer matching [7, 102] and the related Hausdorff distance [70], based on the distance of edge images. Figure 4.4 presents the method for computing the distance of two edge images. It can be parameterized both by the sensitivity of the underlying edge detector and by the level of "forgiveness" for mismatches,
- mutual information [111, 138, 156].

It is worth noting that values acquired from different functions are bound to remain incomparable. Also, while some of these metric propositions have a linear time computation complexity in terms of image size, others are much more complex. Because of this, in cases where the final FSA algorithm is meant to perform real-time not all metrics might prove to be feasible.

A number of properties of the considered metrics can be utilized during the implementation, to improve the efficiency of the fFSA approach. First, by using the symmetry of metrics we can avoid computing the distance twice for a given pair of frames. This optimization has been implemented for testing the fFSA approach and is used by default, regardless of the choice of a particular metric. Secondly, multiple of the discussed metrics can be divided into two steps – the images undergo a kind of preprocessing (e.g. histogram computation) before the processed vectors are compared. To prevent multiple repetitions of the preprocessing for a frame (on each comparison with other ones), the processed vectors are created in a single pass, before the fFSA step is run. This approach is encouraged in the FSA testing environment and newly implemented metrics can easily define such a separation.

4.3.4 Decision rule – A, λ

The decision rule of an FSA method is a function responsible for establishing the current frame's final classification. Because of the broad range of possible decision rule algorithms, they may be provided in a textual, descriptive form or pseudocode rather than a simple formula. The set of possible rules includes but is not limited to:

- acclamation/majority/threshold criteria – the decision for the current frame changed if the respective **number/ratio** of frames in the neighborhood is different;
- weight distribution based on frame distance and/or similarity and/or classification quality - the decision for the current frame is changed according to a weighted sum of decisions in the neighborhood. The weights are adjusted according to the temporal proximity of each frame and/or its similarity to the current frame. For OFA algorithms which provide information about the certainty of a classification (e.g. neural-network based) the weights may further be adjusted with it;
- dynamics of the time-series - treating the classification sequence as a discrete representation of a continuous function of time, we can analyze the dynamics of change in the time series by limiting the change in derivatives of any order;
- approximating the time sequence in the neighborhood with a smoothing function and assigning the approximated value to the current frame;
- Bayesian reasoning based on the distributions learned from training data;
- answering with a value supporting the assumed structure of scene distribution - e.g. not more than a single scene change within a single window;
- identifying and inverting the most outstanding outliers.

In the previous chapter, Figure 3.10 contained an illustration of possible additional information included in the decision rule's reasoning. The time distance Δt from the central frame is explicitly defined and the similarity distances $d(\cdot, \cdot)$ between frames can be computed from the sequence of frames F. Thus, only a way of including the classification confidence into the decision remains to be specified. For this, let us note that most of the proposed decision rule approaches also apply to sequences with real values from the $[0; 1]$ range. The sequence of OFA classifications O can be combined with the classification confidence values K into a new sequence O^K which could be processed by those decision rules. In such a sequence, O^K_m corresponds to the probability of the ground truth value being equal to 1, i.e. $O^K_m \approx c$ when the OFA algorithm strongly indicates a c answer ($c \in \{0, 1\}$) and $O^K_m \approx 0.5$ when the result is virtually undetermined. The OFA algorithms used for the experiments in this work don't return a classification confidence value, therefore using O^K in the FSA step will be noted in Chapter 7 as a direction for future work.

The decision functions return either an exact value of 0 or 1, or are allowed to return a real value in the range of $[0; 1]$. In the latter case, the decision is completed by the application of an acceptance threshold A. The acceptance threshold for the frame at position m and its decision

value is applied as follows:

$$C_m = \begin{cases} 1 & \text{if decision} > A \\ 0 & \text{if decision} < 1 - A \\ O_m & \text{otherwise} \end{cases}$$

To preserve a unified notation for all decision functions, for those returning only 0 or 1, we apply an artificial implicit threshold of 0.5.

The behavior of many possible decision functions can be specified by introducing more controlling parameters. Such functions implemented within this work have a single additional parameter λ, which controls the distribution of frame relevance in the window.

4.4 Computational complexity

In the case of the OFA algorithms, the computational complexity is a simple factor of the number of frames and the complexity of classifying a single frame. The latter is a function of the image size. The complexity of a given OFA algorithm OFA_x can therefore be expressed as:

$$\mathrm{OFA}_x \sim O(n \cdot f_{\mathrm{OFA}_x}(D))$$

where D - image parameters, $f_{\mathrm{OFA}_x}(D)$ - function expressing the complexity of classifying a single image with the given parameters.

The considered FSA algorithms operate on a sequence of n frames with binary classifications, using a shifting window of given size w,

In the proposed form, the FSA step's time complexity for the iFSA variant can therefore be expressed as $\sim O(nw)$, since the processing of a single shifting window is linear in terms of its size (for the methods considered in this work).

The fFSA complexity requires to count in the time for comparing all frames in the window with the central frame. The complexity of comparing two frames of size D depends on the particular metric d and it is expressed as $\sim O\left(nwf_d(D)\right)$

The total complexity of classifying the video sequence is therefore the composition of OFA_x and the complexity of the FSA step. In the case of iFSA, this results in:

$$\mathrm{OFA}_x + \mathrm{iFSA}_w \sim O\left(n \cdot f_{\mathrm{OFA}_x}(D) + n \cdot w\right) = O\left(n(f_{\mathrm{OFA}_x}(D) + w)\right)$$

and for fFSA:

$$\mathrm{OFA}_x + \mathrm{fFSA}_{w,d} \sim O\left(n \cdot f_{\mathrm{OFA}_x}(D) + n \cdot w + nwf_d(D)\right) = O\left(n(f_{\mathrm{OFA}_x}(D) + wf_d(D))\right)$$

The value of w is limited by the necessity of keeping the frames within one window similar with each other. This maximal value $w_{\max}$ is specified for any given category of video. Also the image size D can be considered limited, as in each domain there is an intrinsic limit on the

image parameters (resolution and image depth) – depending on technical possibilities of image acquisition and transmission.

If we consider $w \sim O(1)$, then asymptotically the complexity of the iFSA algorithms is not any bigger than its underlying OFA's (which is also reflected by the minor overhead). It is preferred not to consider a similar reduction for D though, both due to the unknown complexity of the $f_d(D)$ function and to underline the big constant otherwise hidden by the big O notation.

4.5 Limitations and exclusions

The proposed FSA approach, based on the idea of processing the classifications within shifting windows, contains a number of inherent limitations:

1. The FSA input in each step is limited to the range of the shifting window - not allowing to consider the full sequence of classifications and video frames for rationalization. A number of other methods (such as HMMs) process the whole sequence of changing states, producing the appropriate outputs. On the other hand, this limitation allows for real-time implementations of FSA methods as well as for the parallelization of processing.
2. As already discussed, the shifting window approach is susceptible to confusion in regions of scene transitions and/or low OFA output confidence. This has been factored in by limiting the FSA methods to continuous video segments of the original input videos .
3. The FSA methods do not necessarily converge - repeated applications of the same FSA method might either loop switching the result in every iteration, constantly change in an undefined manner or keep flatting out the result to a constant line. A deeper analysis considering the thresholds conditioning change and/or the initial variability of the output might be necessary.

Furthermore, due to the wide scope of available formulations of FSA methods, we have to select a possibly broad but also limited scope of algorithms for further analysis and experimentation. The selected iFSA and fFSA algorithms will be presented further on in section 5.5.

4.6 Extension into multi-categorical classification

While this work focuses on binary classifications, it is important to note that the discussed methods can also be extended into multi-categorical classifications. In this section, one exemplary extension will be discussed on a simple case. Let us consider a problem of assigning sceneries to pictures:

$$\mathcal{C} = \{\mathsf{beach}, \mathsf{forest}, \mathsf{mountains}\}\,.$$

The class assigned by the algorithm can be represented in a number of forms. If we decided to use consecutive numerical values (1-beach, 2-forest, 3-mountains), some of the commonly used classifiers (e.g. neural networks) could reflect the relations between the particular numerical

values. For example, forest would be the mean value of beach and mountains ($2 = \frac{1+3}{2}$), or mountains equal to thrice a beach ($3 = 3 \cdot 1$).

A common method of representing the output of a classifier returning one of k labels is the *1-of-n* encoding ([117]; also called the *One-Hot encoding*). In this type of encoding, instead of a single integer value $i \in [1 \dots k]$, the category is represented by a vector $i^* = (\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{k-i})$. Assigning a vector of k binary labels to each classification ([100]-beach, [010]-forest, [001]-mountains) makes the classifications orthogonal and prevents introducing unexpected relations. In the case of neural networks this encoding can be implemented by having k neurons in the output layer.

When the 1-of-n encoding is used, the classifier returns k values from $[0; 1]$. The returned values are interpreted as classification confidence levels (possibly – as probabilities, if e.g. a softmax layer is used) and the category corresponding to the largest one is chosen. Furthermore, the difference between the two highest values can be used as a measure of confidence of the choice. Such a model has been presented in Figure 4.5a, where the classifier κ_{MFB} returns a three-dimensional vector for every frame. The maximal confidence levels for each frame have been emphasized with a bold font.

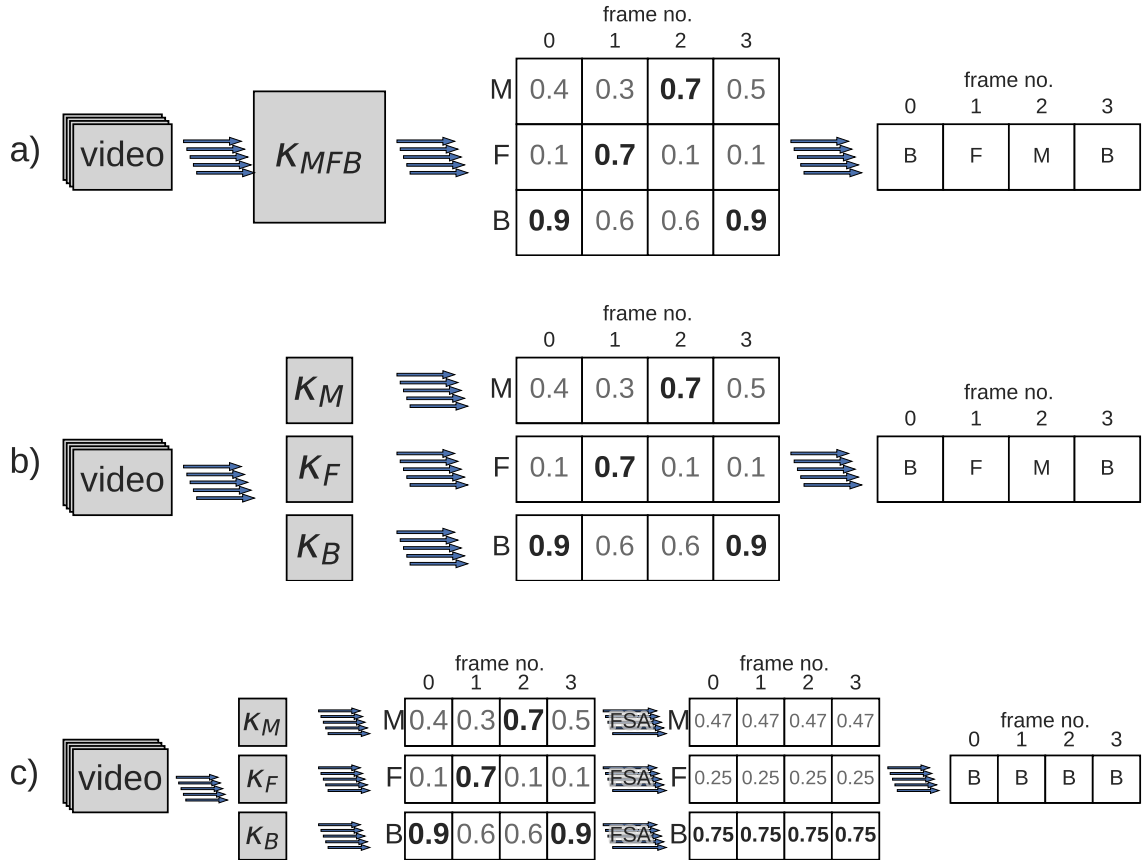


Figure 4.5 – Generalization of binary classification improvements to multi-categorical problems

It is worth noting that κ_{MFB} can be perceived as a composition of three separate binary classifiers – one for each of the output dimensions (κ_M, κ_F, κ_B). Such a model, as presented in Figure 4.5b, does not change the final class assignment, but allows to treat the individual binary classifiers as OFA algorithms. This exposes the possibility to introduce FSA reasoning, by improving the individual outputs' stability, when the continuity of the underlying video is ensured. The preliminary classification results can be substituted with the continuous values ($[0; 1]$) provided by a number of the proposed decision functions. As mentioned in subsection 4.3.4, decision rules can be defined not only for simple binary sequences, but also for those with a provided classification confidence. Furthermore, before the acceptance threshold A is applied, most decision functions return a numerical value from the range $[0; 1]$, which corresponds to classification confidence. These properties allow us to introduce an FSA step to adjust the outputs of each of the considered classifiers before the class assignment. Such an approach has been presented in Figure 4.5c.

Summing up, the binary classification problems considered within this work can be straightforwardly extended into the domain of classification into multiple categories. The discussed approach can be described as top-down: decomposing a multi-categorical classifier into OFA classifiers. Also a bottom-up concept can be proposed, where a number of independent FSA classifiers would be combined into a system classifying frames into multiple categories. In such a composition, each binary classifier would answer the question "does the given image belong to the category number i?" and provide a confidence level for its answer. Those possible extensions will further be noted in the conclusions in Chapter 7 as directions for future work.

4.7 Evaluation criteria

4.7.1 Accuracy

The main criteria for evaluating classification algorithms in works concerning binary classification involve accuracy, sensitivity (True Positive Rate) and specificity (True Negative Rate).[5] For a sequence of classifications $C_{m\in\mathcal{M}}$ with the ground truth $G_{m\in\mathcal{M}}$ (where $\mathcal{M}$ is the sequence of integer indexes of the frames in the video) the classification quality metrics are defined by the ratio of set cardinalities:

$$\text{Accuracy} = \frac{\#\{m \in \mathcal{M} : C_m = G_m\}}{\#\mathcal{M}} \tag{4.2}$$

$$\text{Sensitivity} = \text{TPR} = \frac{\#\{m \in \mathcal{M} : C_m = G_m = 1\}}{\#\{m \in \mathcal{M} : G_m = 1\}} \tag{4.3}$$

$$\text{Specificity} = \text{TNR} = \frac{\#\{m \in \mathcal{M} : C_m = G_m = 0\}}{\#\{m \in \mathcal{M} : G_m = 0\}} \tag{4.4}$$

[5] In many cases the Precision/Recall pair is used instead, which does not reflect the amount of True Negative detections, and is therefore bound to the semantics of a positive/negative classification.

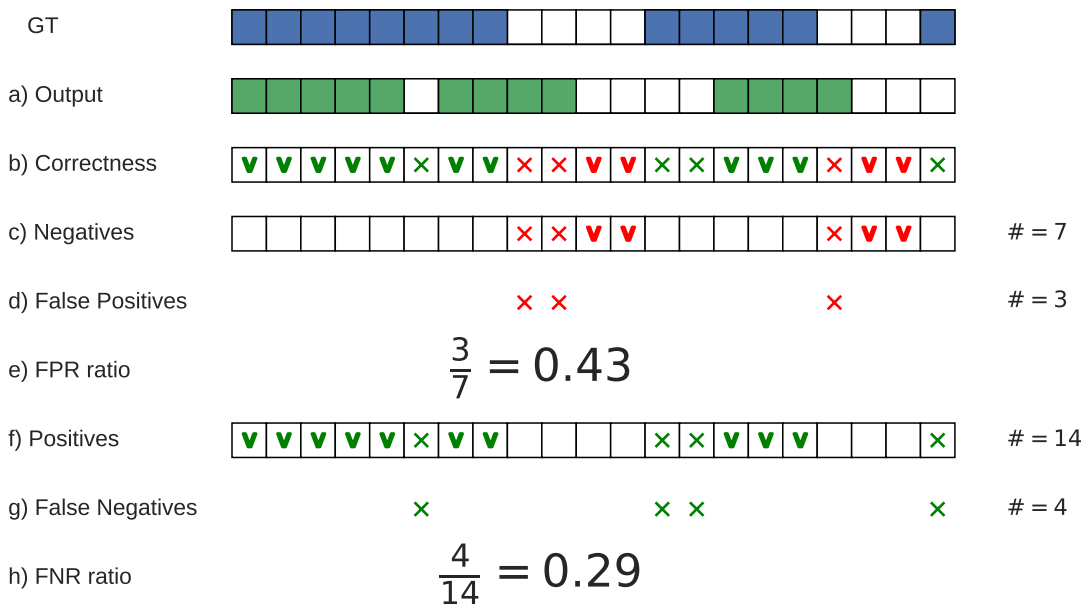


Figure 4.6 – Exemplary evaluation of the FPR and FNR measures.

Since the goal of this work is to improve established methods which already have a very good performance (where published works acquire scores of 70%-100%), we intend to reduce the number of mistakes made by them. Therefore the complementary measures of False Positive Rate and False Negative Rate are of more interest:

$$\text{FNR} = 1 - \text{Sensitivity} = 1 - \text{TPR} = \frac{\#\{m \in \mathcal{M} : C_m = 0 \wedge G_m = 1\}}{\#\{m \in \mathcal{M} : G_m = 1\}} \tag{4.5}$$

$$\text{FPR} = 1 - \text{Specificity} = 1 - \text{TNR} = \frac{\#\{m \in \mathcal{M} : C_m = 1 \wedge G_m = 0\}}{\#\{m \in \mathcal{M} : G_m = 0\}} \tag{4.6}$$

An exemplary evaluation of those measures has been presented in Figure 4.6:

a) The FPR/FNR measures are evaluated given an algorithm output for a video with a known correct classification (GT, the ground truth).
b) Knowing the ground truth, we can evaluate the correctness of the given input. The correctness markers carry two values: if their corresponding output value is correct (x or v symbol) and the ground truth value (green/red color). Therefore the following meaning can be attached to the markers: v-true positive; x-false positive; v-true negative; x-false negative.

For the FPR measure, the correctness of classification of values false in the GT is considered:

c) frames with negative values in the GT,

d) frames incorrectly classified as positive,
e) the ratio of false positives (d) to negative frames (c).

The measure FNR is symmetrical, counting the incorrectly classified positive frames:

f) frames with positive values in the GT,
g) frames incorrectly classified as negative,
h) the ratio of false negatives (g) to positive frames (f).

4.7.2 Stability

Another measure for the classification of sequences involves an evaluation of the quality of the acquired classification sequences. For tasks such as video summarization and scene segmentation the quality is measured by scene overlaps or the placement of scene divisions. Possible mistakes in this matter include scene divisions not being placed where they should be (resulting in missing information in the summarization) or being mistakenly put into wrong places (resulting in a unnecessarily longer summarization).

Multiple works have proposed methods of evaluating the quality of video segmentations. [153] introduced measures of coverage and overlap for evaluating Logical Story Unit (scene) segmentations. [20] used those metrics in addition to an F_1 score which combined precision and recall, with a 20 second tolerance for frame boundaries. [135] proposed a number of methods of evaluating the video scene/shot segmentation quality. [68] addressed the problem of performance evaluation of video scene detection algorithms by defining the False Positive/Negative Indices as well as the Overall Performance Index. Similar to the other methods, the basis for the computation are the numbers of missed and incorrect detected scene boundaries.

The simplest approach is the one presented in [55] which used a simple precision/recall evaluation of a number of tested shot segmentation methods. The tested algorithms' output events (shot boundaries) were considered correct as long as there was a detection in the range of a given threshold.

In the case of OFA and FSA algorithms, the main goal is not to just acquire scene divisions, but also to get the proper transitions from positive to negative scenes. Therefore, a detected scene transition can be counted as a match only if it is a transition in the right direction (positive→negative or negative→positive).

Taking the aforementioned methods and restrictions into consideration, an approach for evaluating the quality of scene segmentations emerges. We assume that given are the actual (Ground Truth) and detected sequences of scene boundaries, as well as a threshold value for accepted distance between matches (B). We will say that two boundaries *match*, if:

- the difference in their time is within the given tolerance B,
- they correspond to the same kind of transition.

If a boundary in the ground truth cannot be matched with a boundary in the evaluated output within the given tolerance B, we will call it a **missing** boundary. On the other hand, if

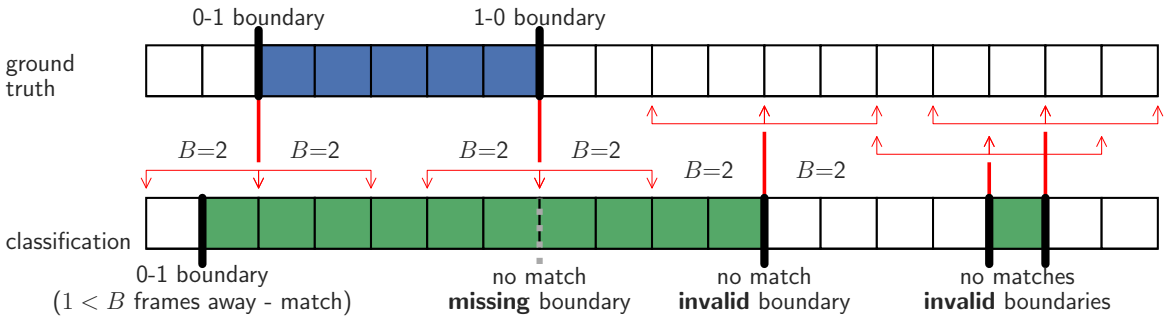


Figure 4.7 – Scene boundary tolerance example for $B = 2$.

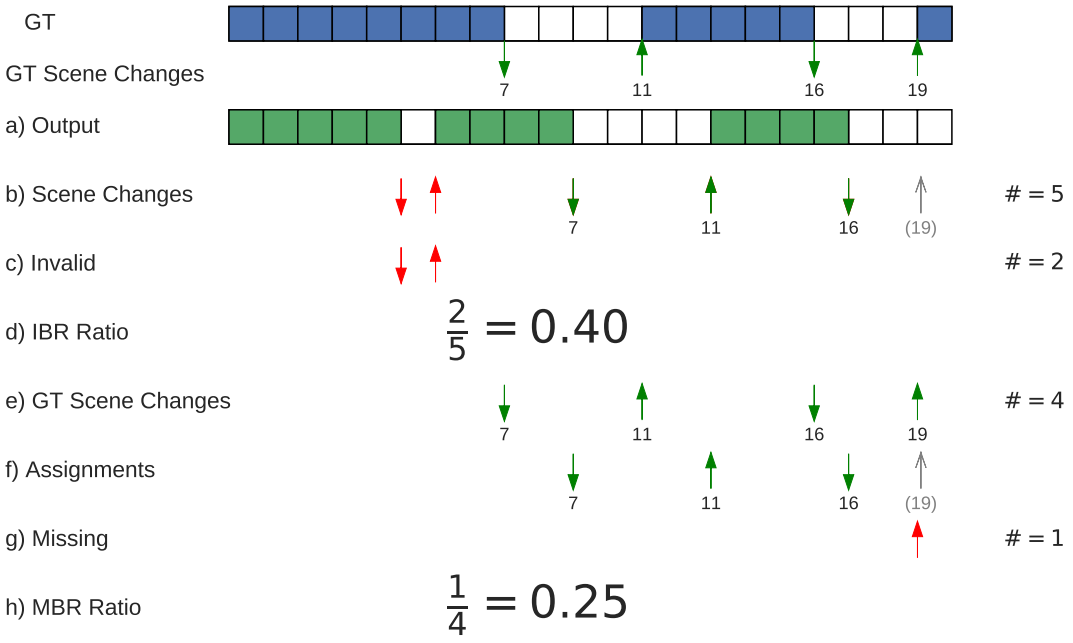


Figure 4.8 – Exemplary evaluation of the IBR and MBR measures for $B = 2$.

a boundary is present only in an algorithm's output but not in the ground truth, we will call it **invalid**. Examples of scene boundary matches have been presented in Figure 4.7.

For the evaluation, the boundaries from the two sequences are paired so that the number of acquired matches is maximal[6].

Out of this pairing we acquire two new measures, defined as follows:

- the Missed Boundaries Rate (MBR_B)

$$\mathrm{MBR}_B = \frac{\#(\text{Missed Scene Boundaries with tolerance } B)}{\#(\text{Scene boundaries in Ground Truth})} \tag{4.7}$$

[6]The pairing can be acquired with a straightforward greedy algorithm.

- the Incorrect Boundaries Rate (IBR_B)

$$\mathrm{IBR}_B = \frac{\#(\text{Incorrect Scene Boundaries with tolerance } B)}{\#(\text{Detected scene boundaries})} \tag{4.8}$$

The evaluation of those measures has been presented in Figure 4.8. The IBR and MBR measures consider the GT and algorithm output (a) sequences in terms of scene changes - distinguishing between positive to negative and negative to positive changes. They provide an analogue of the FPR/FNR approach in the domain of scene segmentation. Because of the ambiguity of scene changes in real-life videos, leniency is allowed and scene changes can be assigned as corresponding if they are in the range of a parameter B from each other. In this figure, for $B = 2$ we compute the IBR as follows:

b) all matched (green) and invalid (red) scene changes in the output,
c) invalid scene changes from the output,
d) ratio of invalid scenes changes to all scene changes in output.

Again, the MBR measure is complementary - counting the scene changes in the ground truth which stayed unassigned:

e) scene changes in the ground truth with their positions in the video (0-based index of preceding frame),
f) scene changes in the algorithm output assigned to GT scene changes (green) and the missing ones (gray),
g) ratio of unassigned scenes to all scene changes in ground truth.

4.7.3 Evaluation overview

In this subsection different scenarios of evaluating classification sequences are presented. The goal is to show that the values of metrics can change independently and that each of them measures a different quality.

Exemplary evaluation results have been presented in Figure 4.9. A classification sequence equal to the ground truth is presented in image a). All measures are equal to 0, since no kinds of errors are present. The consecutive sequences present exemplary output results with their evaluations (correctness of single classifications, scene divisions and their matchings, missing scene divisions).

Scenarios b) and c) contain exemplary values related to the misclassification of whole scenes. Since the classification output is binary, a missing scene results in longer intervals between scene divisions, therefore influencing the MBR value.

In the case of d), a number of random isolated mistakes is introduced. The measure influenced the most in this case is IBR, since every mistake in a uniform sequence adds an artificial invalid scene division.

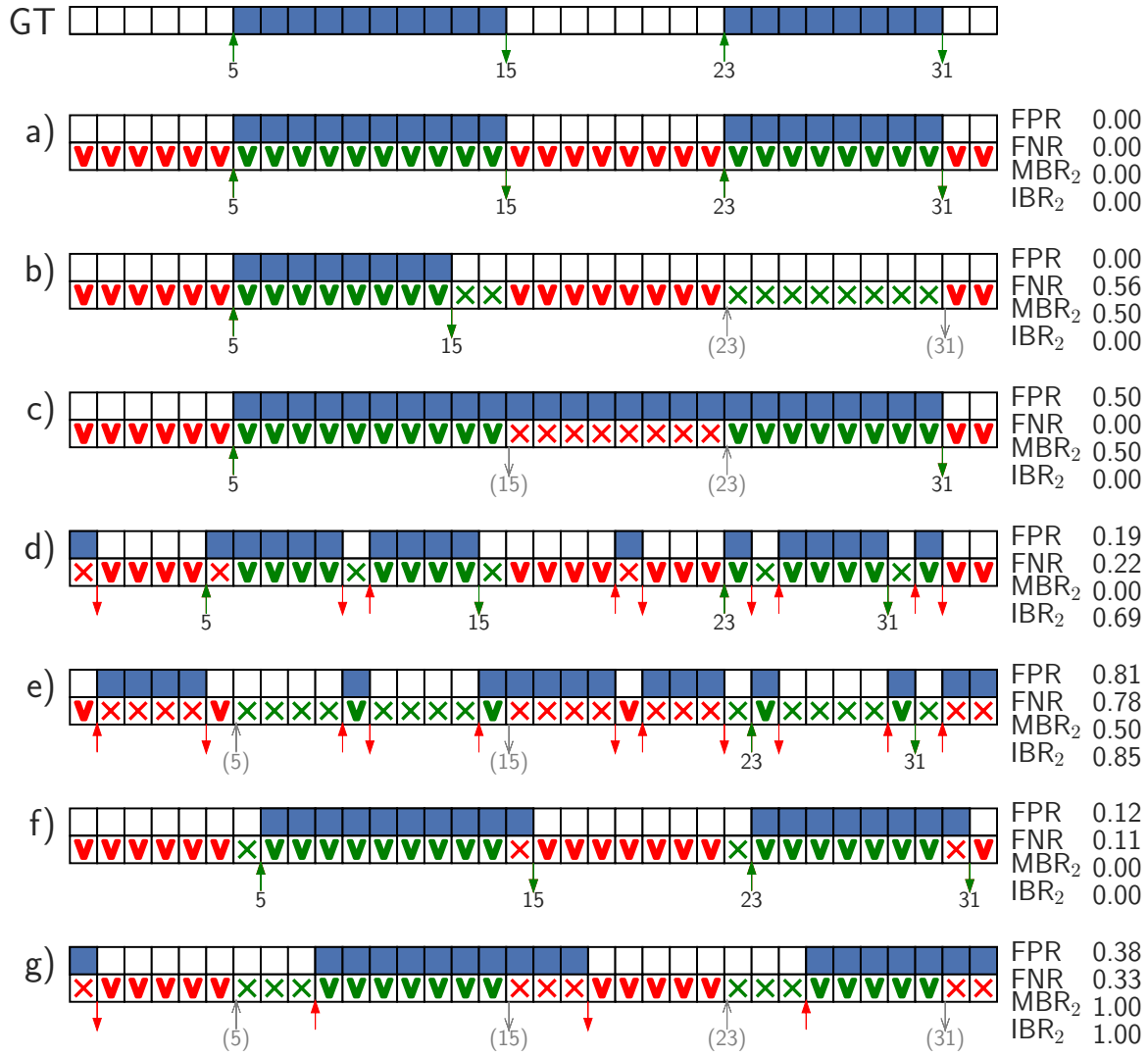


Figure 4.9 – Examples of FPR/FNR/MBR/IBR metric values for different evaluated sequences. GT - Ground truth; a) exact match; b-f) Exemplary output sequences.

Example e) is a full inversion of the classifications from d) - true values have been changed to false and vice versa. It can be seen that the corresponding accuracy measures sum up to 1.0, since the classifications change accordingly. At the same time, due to the scene change tolerance the stability measures might not be as influenced by a complete change of classifications.

Examples f) and g) show the importance of a proper choice of a tolerance limit. For a limit of 2 frames distance, the scenes shifted by one frame from the ground truth positions, still acquire the best possible stability measures. A shift exceeding the limit might lead even to a complete scene boundary mismatch, as in g).

4.7.4 Combining measures

When two algorithms are compared in terms of more than one measure at once, the only natural ordering of results is a *partial order*. Two results are equal if all their corresponding measure values are equal. One result is better than another if it is better in terms of at least one measure and better or equal in terms of all other ones.

Such an ordering corresponds to an intuitive understanding of the order between results. But also, not all results are directly comparable with each other. When one solution is better in terms of one measure and a second one is better in terms of another measure – they remain incomparable in terms of the partial order.

To provide a possibility to select a best result out of multiple ones, we need to introduce a method which allows to express our understanding of the results' preference. A typical approach in such cases is to provide either a function which combines multiple criteria into a single value (e.g. the F_1 score if the algorithm is evaluated in terms of precision and recall) or provides a generalized difference of measure vectors (e.g. a lexicographical comparison).

Contrary to the partial order, the choice of such a function can be arbitrary. For example, each kind of error can have a domain-specific cost assigned to it – in medical applications false negatives can lead to undetected diseases, while false positives only cause costs related to additional diagnostics or unnecessary treatment. In such a situation, the cost (weight) of false negative errors would be significantly higher.

When multiple measures are considered, the comparison of algorithms can also be blurred if two measures are strongly related with each other. The decrease of one kind of errors might be strictly related with the increase of errors of the second type. In binary classification algorithms these relations are often presented as the ROC (Receiver Operating Characteristic) curve or Precision-Recall curve. The shift from one error to another is usually controlled by a threshold parameter. Two results incomparable in terms of the partial order may therefore correspond to the same method with a different value of the positive threshold parameter.

Within the experiments in this dissertation we need to combine sets of measures mainly in two scenarios. First, when the proposed FSA method parameterization is optimized for a dataset and we need to select the parameter set which performs best. Secondly, when two particular results are compared with each other and a descriptive evaluation of their difference is needed.

Rank aggregation

The two pairs of measures (FPR/FNR and IBR/MBR) have values in the same ranges ([0;1]) and are defined in a similar manner. Nevertheless, their values represent different concepts with different distributions. A straightforward combination of them (e.g. average) could therefore be questioned, because it is an implicit and arbitrary assignment of equal cost weights for each error measure unit.

One universal approach in such cases is the application of rank aggregation methods. Nu-

merous approaches have been proposed in this field [39, 95], with the goal of determining the consistency of preference across multiple evaluations of the same entities. Practical applications range from comparing product rankings, creating meta-reviews of research results, combining search results from multiple engines. The application of ranks allows to combine results expressed in incomparable units and compute statistics which do not require strong assumptions about the distribution of the considered values.

The experiments described in Chapter 5 require choosing the best set of parameters evaluated in terms of three measures (FPR, FNR, IBR). In this case, the best results are selected by comparing them in terms of their average ranks. The ranks were created using the so called *dense* ranking, i.e. equal values share a single rank and the values following them receive the next rank number[7].

At this point it is also important to note some pitfalls of such an approach. First, methods based on rank aggregation require a sufficient number of compared records and/or considered measures to provide meaningful or statistically significant results. Furthermore, the comparison of two records can yield different results already for slight changes in the population which was used to compute the ranks. And finally, to provide a descriptive explanation of the results and account for a possible shift on the ROC curve, the actual values of different measures might be more practical.

Symmetrical measures

Due to the aforementioned issues with the rank aggregation approach, an additional method for comparing records of error measures is proposed for more detailed analyses of particular results.

The strong relation of the FPR-FNR measure pair can hinder the analysis of such strong effects as those introduced by distortions. For example, if a distortion of the input video caused a 3% decrease of the FPR and a 30% increase of the FNR, the overall result could be intuitively described as worse than that without the distortion. At the same time, in terms of the partial order the pairs of values are not comparable and for rank aggregation a broader context is needed.

Works which express the results in terms of ratios of correct findings (TPR/TNR or Precision/Recall) usually propose combined measures such as geometric, harmonic, or arithmetic means. The commonly used accuracy measure is a weighted arithmetic mean of TPR and TNR, where the weights are determined by the ratios of positive and negative samples. The aforementioned F_1 score is a generalized harmonic mean of precision and recall. Comparing to the arithmetic mean, it leans stronger towards the smaller of the averaged values.

We see that in some situations it is not possible to provide meaningful results or interpretations of multiple measures or to apply an evaluation based on rank aggregation. A single score, which can be evaluated from a a single record with quality measures (without the whole population being available) is required. For the comparison of pairs of related measures a combined value

[7] This ranking is also referred to as the "1223" method.

will therefore be considered where necessary – the root mean square (RMS):

$$\mathrm{RMS} = \sqrt{\frac{\mathrm{FPR}^2 + \mathrm{FNR}^2}{2}}. \tag{4.9}$$

The RMS has been chosen due to the following properties:

- It is consistent with the partial order (i.e. if two results are comparable in the partial order, their RMS comparison result is the same).
- The RMS is a mean which shows a preference for the higher of the values. This allows to make stronger claims about the acquired results, because the error rate of the algorithm will not be underestimated.
- For FPR and FNR values lower than 50%, the difference between the RMS value and an averaged error estimate acquired by computing a harmonic mean of TPR and TNR is very small. It lies in the range of $(-0.5\%;\ 3.3\%)$. This shows that while its arithmetic derivation is more straightforward, its value is close to that of other measures.

Using the RMS to combine pairs of symmetrical measures is not free from the issues which require us to use rank aggregation for the selection of optimal results. Therefore, its usage will be limited only to the necessary cases.

Previous solution - L^2 norm

This option is not used anymore in the experiments in this dissertation. It is noted for the completeness of description and to relate to papers connected with this work which have used it.

In the of publications [12–15], whose preparation and publication were interleaved with those of this dissertation, the L^2 norm was used to connect the three error measures. The concerns related to such an approach have been noted above, in the description of the rank aggregation approach. In this context, it is important to note that regardless of the possible pitfalls, thus far the L^2 norm performed properly for the currently considered datasets and algorithms.

In the considered cases, the improvement ratios turn out to have relatively similar distributions. Therefore, it did not occur that just one of the measures strongly dominated the other ones when the selection of an optimal result was performed. Actually, in the considered cases the L^2 norm acquired often a more uniform improvement of all measures than rank aggregation. Furthermore, it is easier to apply in terms of the implementation and computational cost, as results are instantly comparable (without the need to compute ranks for the whole population) and partial executions of the experiments remain informative (by retaining the order of the considered records).

The L^2 norm has been replaced by the rank aggregation method after a reviewer noted that its applicability is not universally justified. This remark has been acknowledged, because although currently the L^2 norm performs very well, such a performance cannot be guaranteed for all future evaluations. Appropriate modifications have been made accordingly.

5 Testing procedure

In this chapter we present the full procedure of testing FSA algorithms. First, we demonstrate the general testing procedure and its main components. Next, the input data and base OFA algorithms are described. After briefly discussing the testing environment, we evaluate a number of candidate methods to select those which are the most promising. Finally we provide the full testing procedure and evaluated sets of parameters.

5.1 Testing procedure outline

The whole testing procedure is executed for multiple problem definitions. Each of them consists of:

1. its detection goal and video data type,
2. the underlying OFA algorithm (black-box),
3. the problem's evaluation criteria (B).

The first two points from the list above will be described in greater detail in this chapter, in section 5.2. The evaluation criteria parameter B has already been presented in the previous chapter, in subsection 4.7.2.

For each of the problem definitions a number of algorithms is evaluated, which are defined by:

1. the tested FSA scheme,
2. FSA-related parameters and the selection of composite functions.

The general outline of the evaluation of a single FSA algorithm instance has been presented in Figure 5.1. The input data is used to acquire the OFA classifications and later, together with them, to run the FSA algorithm - resulting in the second sequence of classifications. The performance of the FSA method is compared with the performance of the underlying OFA method, resulting in information about the improvement of classification quality according to the four criteria (FPR, FNR, IBR, MBR). The dotted arrows and box represent the part relevant only for the fFSA variant of the proposed method. When the attributes of a particular fFSA execution are known, the inter-frame similarities can be computed and used as an additional input for the reasoning in the FSA step.

Figure 5.2 presents a diagram of the full set of parameters for the testing procedure of an FSA algorithm. It is worth noting, that the parameters are significantly connected, therefore the full experimentation routine will not involve a complete Cartesian product of all possible values.

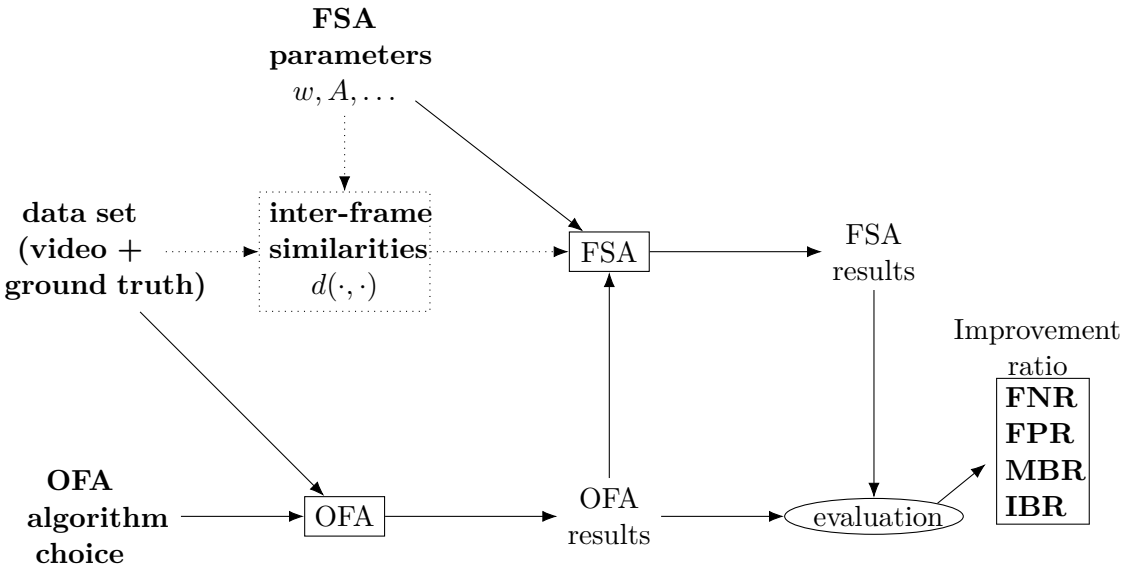


Figure 5.1 – Procedure for evaluating OFA and FSA classifications.

For instance, depending on the domain, videos are available only in particular resolutions and frame rates. Furthermore, not all OFA algorithms provide information about the classification confidence and not all algorithms are able to utilize information coming from the parameter functions.

A more specific instance of the diagram has been presented in Figure 5.3. It illustrates the scope of tests performed within this dissertation. The decision functions for the FSA method are already known. So are the particular parameters of the FSA scheme, which also have their ranges specified. The details regarding the considered ranges of parameters are described in later parts of this work as well as in [12]. The test data has also been chosen and particular datasets are listed. Their detailed description will further be provided in section 5.2.

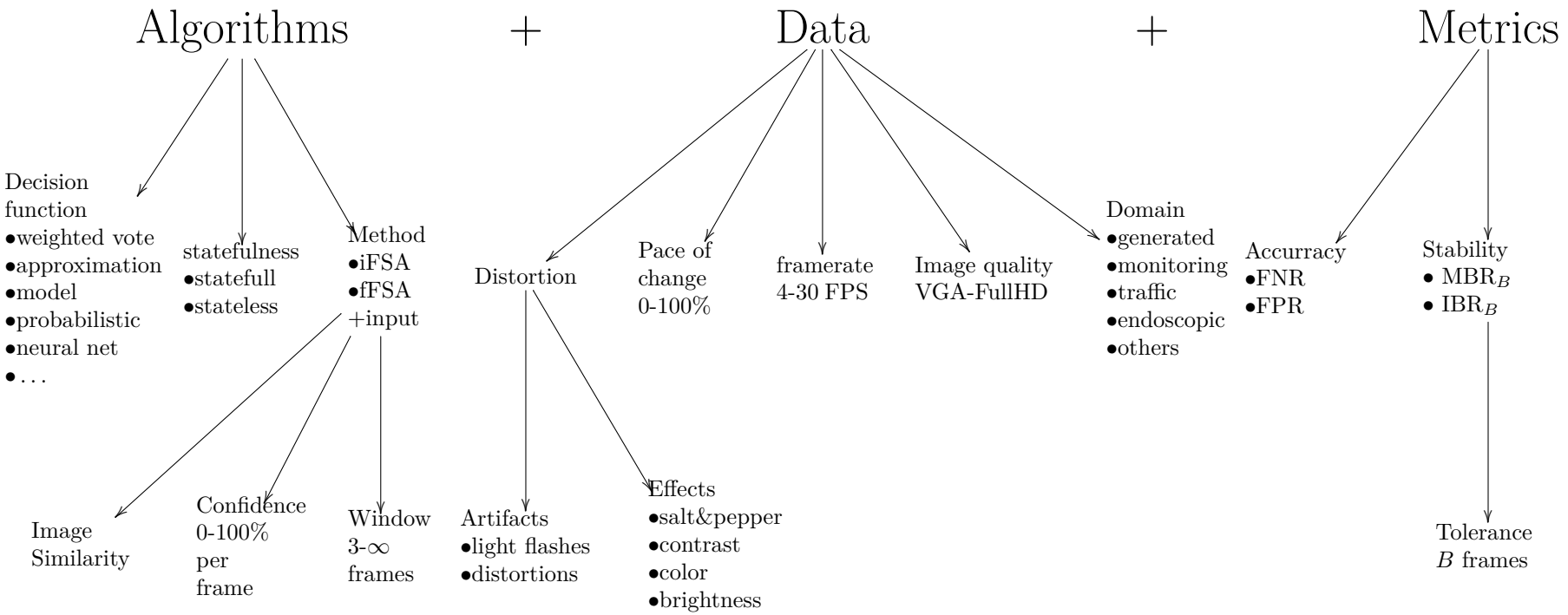


Figure 5.2 – The testing parameters. Different FSA algorithms are tested against the given data and calibration/charateristic parameters.

Figure 5.3 – The testing parameters - for the tests performed within this work.

5.2 Test data

Table 5.1 – Overview of test datasets and their corresponding OFA algorithms.

Dataset	**Movers**	**Chokepoint**	**Traffic**	***Endoscopic***
Resolution	1680 x 1050	800 x 600	640 x 480	720 x 576
Color depth	24bit	24bit	24bit	24bit
Frame rate	25FPS	25FPS	25FPS	25FPS
Total frame count	unlimited	Portal 1: 25K Portal 2: 34K (each × 3 cameras)	11179	>1M
Frames in experiments	20K	20K (× 2 portals)	11179	—
true classification semantics	any silhouette visible	any face visible	a green traffic light visible	lesion or bleeding visible
OFA algorithm	openCV, Haar cascade silhouette detector	openCV, Haar cascade facial detector	Traffic Light Recognition (TLR) [30]	*multiple*
OFA performance				
FPR/FNR	0.01/0.13	Portal 1: 0.08/0.03 Portal 2: 0.24/0.10	0.31/0.21	N/A
MBR/IBR	0.12/0.72	Portal 1: 0.06/0.49 Portal 2: 0.32/0.60	0.00/0.99 (see also in subsection 5.2.3 and Table 5.2)	N/A

The experiments performed for evaluating the FSA method involve the usage of three datasets and four OFA algorithms. In the case of the Chokepoint dataset two subsets are considered (Portal 1 and 2). A general overview of the datasets and algorithms has been presented in Table 5.1. The endoscopic dataset has been included in the table only for the completeness of the description, as it has not been used in the experiments. The following subsections contain a more detailed discussion.

5.2.1 Artificial video stream (movers)

Data

The artificial video stream is generated video data created with arbitrarily adjustable parameters for performing tests in a fully controlled environment. Its contents include a moving silhouette on a static background. The detection accuracy of the silhouette is influenced by overlapping moving objects, which at times hinder the view (as presented in Figure 5.4). In the experiments it is denoted with the name *movers*.

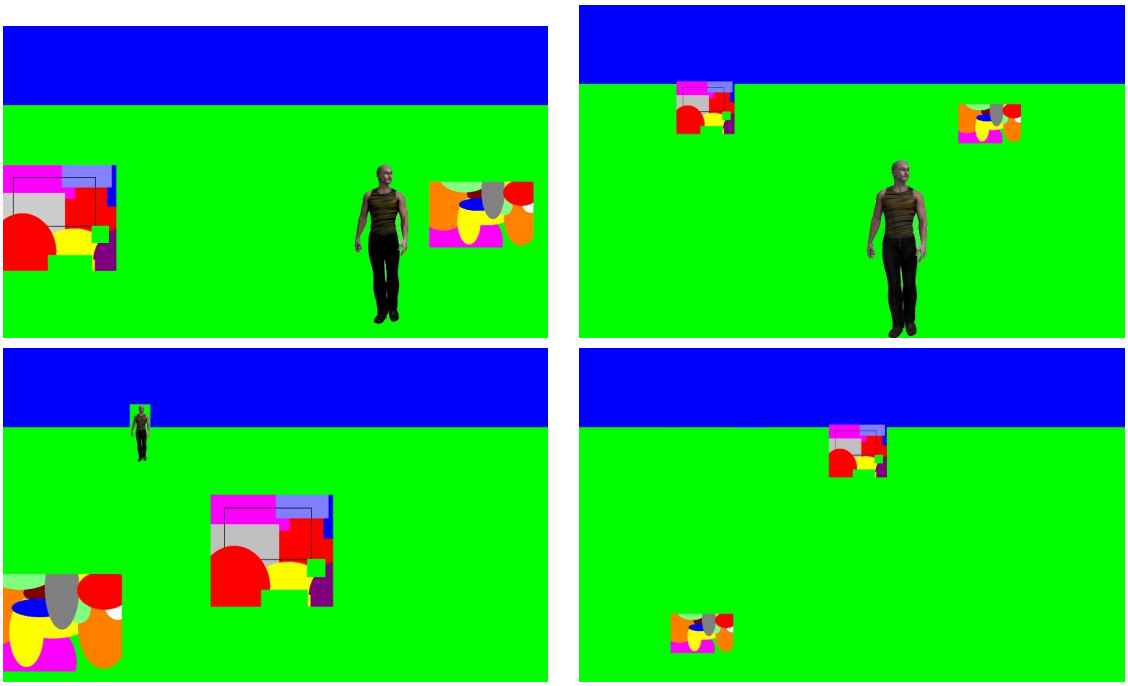


Figure 5.4 – Exemplary images for the movers video stream. The sought object is the silhouette. Other moving objects visible in the pictures, blur, static and dynamic distortions introduce misclassifications with a moderated frequency.

OFA Algorithm

The Haar Cascade Classifier [129, 154] from the OpenCV library [23] has been used with the packaged default silhouette detector. The detector performs with a great accuracy in cases where there is a clear view of the figure, but can be fooled by the overlapping shapes. Furthermore there is no specified definition of the minimal amount of the figure to be visible, at which the detector reports a positive. Therefore the actual performance of the resulting algorithm can vary by a couple of percent point in cases where there is more or less movement around the edges of the view.

Figure 5.5 – Exemplary images for the ChokePoint video stream. Both full (left) and partial (right) visibility of the passing faces is present in the recordings.

5.2.2 Chokepoint recordings

Data

The ChokePoint [164] dataset is a set of annotated surveillance recordings designed for experiments in person identification and recognition by the National ICT Australia Limited (NICTA)[8]. Its license makes it available for non-commercial research purposes [9].

[8] `http://arma.sourceforge.net/chokepoint/`

[9] Choke Point Licence:

This dataset ('Licensed Material') is made available to the scientific community for non-commercial research purposes such as academic research, teaching, scientific publications or personal experimentation. Permission is granted by National ICT Australia Limited (NICTA) to you (the 'Licensee') to use, copy and distribute the Licensed Material in accordance with the following terms and conditions:

1. Licensee must include a reference to NICTA and the following publication in any published work that makes use of the Licensed Material:
   Y. Wong, S. Chen, S. Mau, C. Sanderson, B.C. Lovell Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81-88. IEEE, June 2011.
2. If Licensee alters the content of the Licensed Material or creates any derivative work, Licensee must include in the altered Licensed Material or derivative work prominent notices to ensure that any recipients know that they are not receiving the original Licensed Material.
3. Licensee may not use or distribute the Licensed Material or any derivative work for commercial purposes

The dataset contains multiple recordings of people passing through portals. The ground truth information (positions of faces of people passing through the doors) is given on a per-frame basis. Exemplary images from the ChokePoint dataset can be seen in Figure 5.5.

It has to be noted that the data contains also clear views of people who do not pass the portals guarded by the cameras. Therefore not all possible faces are included in the provided annotations.

OFA Algorithm

The Haar Cascade Classifier [129, 154] from the OpenCV library [23] has been used with the distributed default face detectors. The output of the algorithm is a list of all rectangular findings, assumed to represent one of the views of faces. Therefore the algorithm can be used both for establishing the presence of any face in the video, but also to count the number of visible people. It has to noted though, that the recordings have been made in an artificial environment, therefore most actors walk alone and with a straight/stiff posture.

The final evaluation of the algorithm has been performed with the special profile of the test data taken under consideration, allowing for ignoring detections of faces visible in the video, but not considered in the annotations.

5.2.3 Traffic light recognition

Data

The traffic light recognition [30] dataset is an open annotated dataset [10], containing a long recording from a car driving through a city's streets. The annotations contain information about visible traffic lights (green/yellow/red/ambiguous) and their positions in each frame. The recording contains 23 scenes for green lights (periods of permanent green light visibility or lack of green lights), most of which are very long. The usage of the recordings is permitted for scientific experiments as long as the authors are acknowledged.

OFA Algorithm

The OFA algorithm for the traffic light detection has been presented and evaluated in the aforementioned paper [30], as well as [37]. Its consecutive steps are:

1. Spot Light Detection:

including but not limited to, licensing or selling the Licensed Material or using the Licensed Material for commercial gain.
4. The Licensed Material is provided 'AS IS', without any express or implied warranties. NICTA does not accept any responsibility for errors or omissions in the Licensed Material.
5. This original license notice must be retained in all copies or derivatives of the Licensed Material.
6. All rights not expressly granted to the Licensee are reserved by NICTA.

[10] data created by the Robotics Centre of Mines ParisTech and are publicly available at: `http://www.lara.prd.fr/benchmarks/trafficlightsrecognition`

Figure 5.6 – Exemplary images for the Traffic Lights video stream.

a) detect bright spots,
b) filter detections by shape;

2. Adaptive Template Matching:

a) decompose into structure of smaller shapes,
b) compute confidence of template.

Due to the general description of the algorithm and its extendable structure not needed for evaluating the FSA approach, the presented outline has been partially simplified. Namely, the Adaptive Template Matching has been implemented in a fixed manner, without the modularity presented in the original paper.

The evaluation proposed by the authors of [30] is temporal matching. By its definition, a detection is counted correctly, if a light was detected at least once during its period of visibility. Those measures significantly differ from the ones we have proposed above. A perfect precision/recall temporal matching result can amount to over 99% FNR or 100% IBR. To approach the improvement of the FPR, FNR and IBR measures, looser criteria have been applied in step 2b, so that more findings are reported and the accuracy measure values are closer to each other.

The comparison of the algorithm version from the original paper with that implemented within this dissertation has been presented in Table 5.2 [13]. The actual results of the original

algorithm in terms of the FPR/FNR/IBR measures are estimated from the two papers' results - the possible ranges have been presented in the table. In the case of the IBR measure, the authors themselves indicated that their algorithm acquires the very high precision values by leaving exact scene matches out of account. This information has led to the very high estimation of the IBR value.

The version of the algorithm used in this dissertation as an exemplary OFA algorithm visibly differs from the original algorithm. The difference of priorities results in a difference of chosen error measures to optimize (temporal matching vs. frame/scene-wise evaluation). The different emphasis is reflected by the trade-off between the FPR and FNR values – neither algorithm performs better than the other when all measures are considered.

Table 5.2 – Comparison of the best-effort implementation with the original TLR algorithm.

	Error measures		
Algorithm	FPR	FNR	IBR
Original from [30] (estimated)	1-10%	37-56%	∼**100**%
Implementation for FSA	30.81%	20.95%	99.32%

5.2.4 Endoscopic examinations

*This subsection has been added only for the completeness of the description. The recordings from endoscopic examinations have **not** been used in the main experiments of this dissertation. Nevertheless, a number of preliminary experiments have been carried out on them.*

Data

Experiments and motivations related to analyzing video data from endoscopic examinations have been mentioned multiple times throughout this work. They have been performed in relation with the Mayday Euro 2012 project, which created a multimedia processing platform based on a computational cluster located in the Gdańsk University of Technology [87, 88]. One of its exemplary applications was a system for supporting medical professionals in endoscopic examinations.

In the course of the project a dedicated dataset was created, which was expected to address the lack of available reference datasets in the field [94]. The main part of the dataset consisted of conventional endoscopic examination recordings [33], collected and annotated by the Medical University of Gdańsk. The annotations had been made on a per-frame basis, with multiple possible labels per frame describing the clarity of view, visible lesions and bleedings. Another part of the dataset consisted of WCE recordings, which at the time of the project had a significantly worse quality. The work on methods for processing long WCE recordings considered the recordings from conventional examinations, which were of higher quality. This reflected the

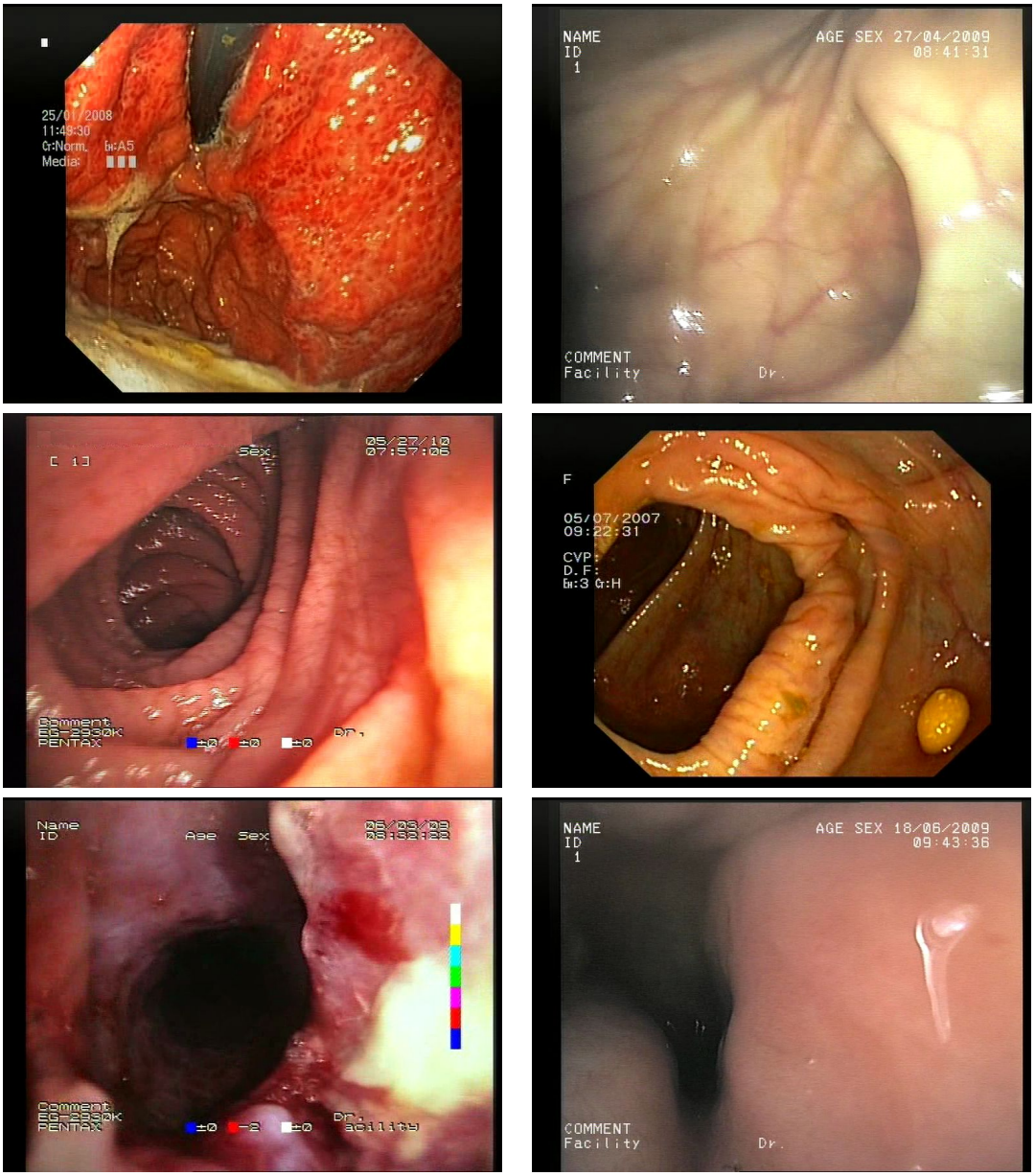


Figure 5.7 – Exemplary images from the Endoscopic Examinations dataset.

anticipated improvement of WCE recordings quality due to the ongoing developments in the field.

Efforts to prepare and publish the recordings with annotations were unsuccessful due to formal reasons (intellectual property). At present their availability is limited.

OFA Algorithms

Numerous algorithms developed for processing WCE recordings [79] were considered for implementation within the Mayday project. A number of them have been evaluated, with a focus

on parallelizing them for a fast analysis of stored long recordings and possible real-time applications [16]. Their classification quality on full videos has raised numerous concerns, because in most cases the classification sequences were heavily segmented, which led to the definition of OFA algorithms and started the considerations for this dissertation.

5.2.5 Distortions

The datasets presented above allow to test the influence of applying the FSA approach to the OFA classification results. To evaluate the robustness of the FSA algorithms to decreasing video stream quality, we will introduce a number of artificial effects on the video streams.

Artificial distortions can be of various types, which correspond to the origin of the real-life distortions they are meant to represent:

- image transformations (blur caused by wrong focus, color changes caused by sudden changes of lighting, motion blur caused by change of camera position),
- random artifacts (random shapes – e.g. light reflections, dirt on the camera lens), and
- dynamic noise - random changes in single frames (salt and pepper, random lines – e.g. damage of old video recordings).

Distortions considered in this work have been divided into two categories according to the expected way in which they are impeding the OFA algorithm and the source of the distortion:

- influencing the **quality** of the recording itself (salt and pepper, some image transformations) and
- **obscuring** parts of the view (artifacts, random lines).

In this simplified model, the quality distortions correspond to internal factors influencing the recording. Those can originate both from a faulty camera or storage media, as well as from improper settings or insufficient parameters of the camera (e.g. lack of focus at a small distance). Also sudden changes in the environment may result in distorted pictures in the time, which the camera needs to adapt. Although the camera's view allows for a perfect picture, the quality distortions hinder taking it.

The obscuring distortions relate to external factors, which are actually present in the view. Those include for example reflections of light, sudden shadows, or rainfall. Even if a camera with a perfect quality was used, those distortions would still be present in the recording. We account for them, because they can strongly influence the classification quality for short periods of time.

Further on, the quality/obscuring distortion distinction will be used.

The effects of the distortions on the images can be expressed either by measuring the relative image quality or by expressing the parameters of the distortions generator. The former is a wide field by itself, where many works attempt to define a way of measuring the perceived image quality (e.g. [161]). The latter depends on the implementation details of a particular distortion

Table 5.3 – Distortion parameters at an intensity of 100%.

Distortion		**Parameter / *additional description***	**Value**
Obscuring	Random lines	number of randomly appearing lines	100
	Artifacts	number of randomly appearing artifacts *small random triangles (x/y span up to $\frac{1}{6}$ of frame's width/height) of random color*	130
Quality	Blur	blur filter kernel size	67
	Salt and pepper	number of pixel color values set to min or max values	50

generator, but also allows for a fine-grained control of the distortion level. And therefore, due to its simplicity and adjustability, further on the the distortion generator intensity will be defined and used to express the distortion of the test video stream.

The levels of different kinds of distortions are adjustable in terms of their intensity. This is understood in multiple ways, as their frequency of appearance or level of change (e.g. blur kernel size or artifacts' counts and sizes). Both distortion categories have their own intensity parameters, which scale the intensity of all of their distortions. Namely, the intensity parameter defines the scale of change of a distortion. We will further express it as a numerical value in a scale of 0-100%.

The appearance frequency and interpretation of the maximal intensity value have been chosen to allow for an interpretable comparison of the test results. First, the appearance frequencies are scaled so that over 99% percent of frames contain at least a minimal distortion. This ensures that the comparison with the original stream does not contain long sequences of undistorted frames. Secondly, the distortion intensities have been scaled so that the level of 100% represents such a value at which images start becoming incomprehensible for a human observer. For each distortion, its intensity is scaled using a parameter of the applied transformation. The values of the parameters corresponding to an intensity of 100% have been presented in Table 5.3. The various distortion types applied to a clear image at exemplary intensities have been presented in Figure 5.8. When a distortion is applied to a given video, its intensity is randomly modulated up to the specified maximal intensity, with an average of 0.75 of the maximal value.

A number of distortions have been chosen for the testing procedure. Their corresponding pairs of quality/obscuring distortion intensity values have been marked in Figure 5.9 and range from a perfect stream to one distorted to a very high level. Applied distortion levels will further on be denoted as such points, i.e. (x_q, y_o) means quality distortions applied with an intensity of x_q and obscuring distortions applied with an intensity of y_o.

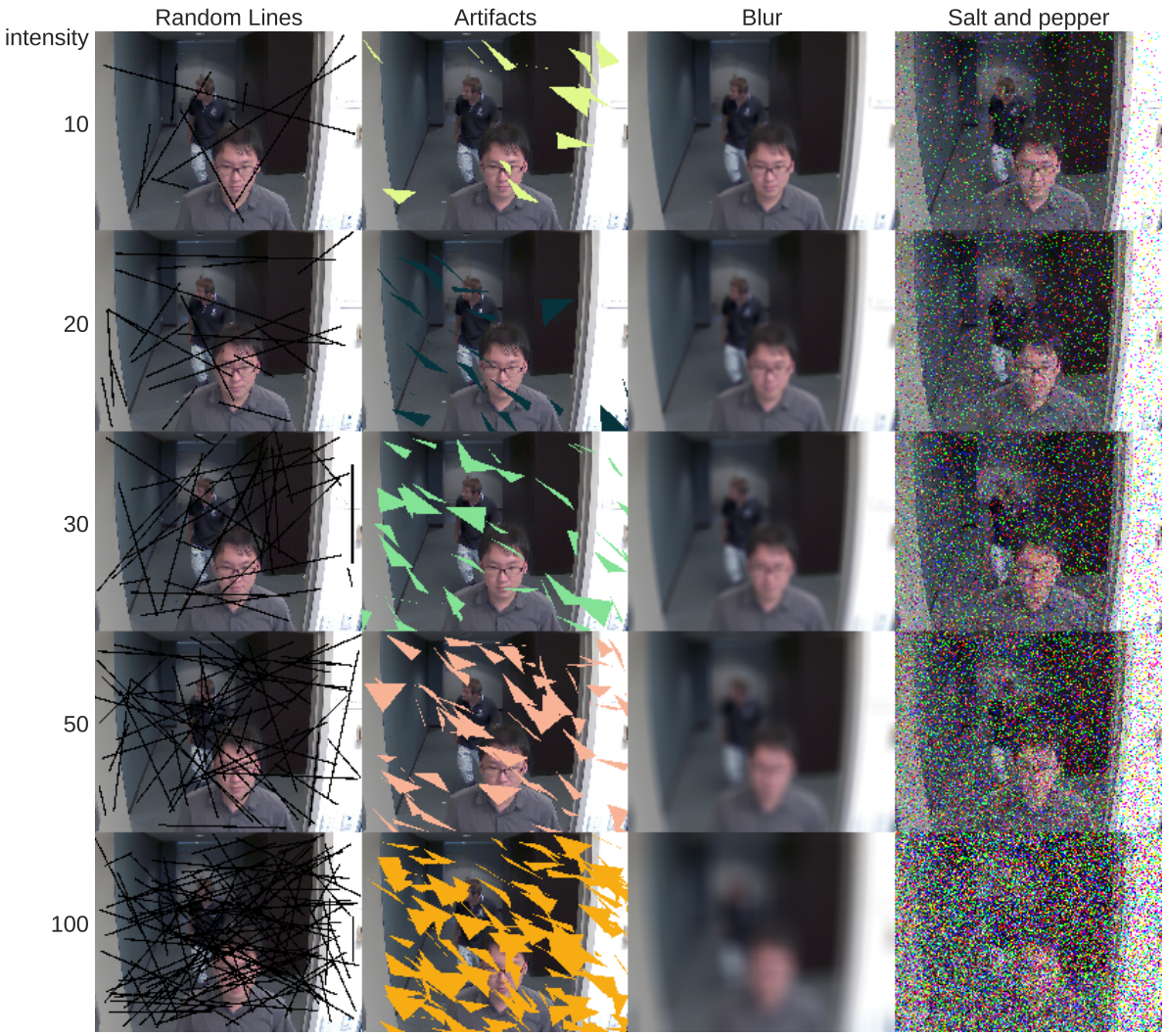


Figure 5.8 – Clear image with various levels of distortion applied.

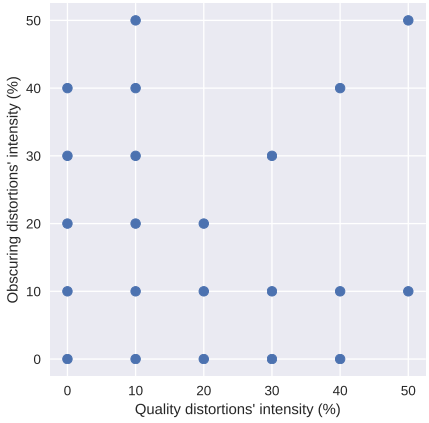


Figure 5.9 – Tested values of quality/obscuring distortion intensity values.

5.3 The testing environment

Since the main consideration of this work is the quality of the classification results and the performance of acquiring them, the specifics of the testing environment don't significantly weigh on the results of the performed experiments. Nevertheless, because floating point numerics and readily distributed implementations of common algorithms are being used, below we provide the necessary specifics of hardware and implementation. As presented, the testing environment is based on the Python built-in modules and set of scientific tools:

- Python 2.7.14,
- numpy 1.13.3,
- scipy 1.0.0,
- scikit-learn 0.19.1,
- ipython 5.4.1,
- Jupyter Notebook 4.4.0,
- matplotlib 2.1.1,
- openCV 3.3.1,
- pandas 0.21.1,
- video and metadata repository - shelve module + file system.

Since Python applications inherit the limitations of floating-point operations from the underlying hardware, it is also necessary to note basic information about the lower layers of the environment.

Parameter	**Value**
operating system	GNU/Linux
distribution	ubuntu
release version	Xenial Xerus 16.04

The computations and final analysis have been performed using the Google Cloud Platform Compute Engine IaaS (Infrastructure as a Service) platform. A dedicated VM has been created, with the following parameters:

Parameter	**Value**
machine type	n1-standard-(processor count), custom
processor count	1-32 (up to 64 available)
hardware/processor	x86_64
RAM	3.75GB - 56GB
disk size	120 GB SSD

It is worth to note that the number of processors and amount of RAM were adjustable – set to higher values during the intensive parallel processing and to lower ones when a manual analysis using the Jupyter Notebook environment was performed.

A general diagram of the testing environment has been presented in Figure 5.10. The algorithm has been visually separated from the runtime environment to underline the loose relationship between them, but the connection stating that relationship remains emphasized.

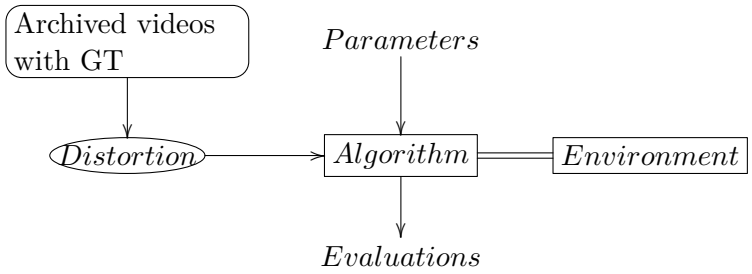


Figure 5.10 – A general diagram of the testing environment.

A more detailed diagram of the actual implementation of the testing system has been presented in Figure 5.11. The design is based on consecutive stages, reflecting the testing procedure:

- the **data set stage** - responsible for generating and converting input datasets to a common format - both the video and ground truth information.
- the **OFA stage** - where datasets acquired from the previous stage are classified with OFA algorithms, what results in creating a corresponding OFA classification sequence for every data set.
- the **FSA stage** - which performs the essential part of the experimentation. Multiple FSA approaches are applied with multiple parameter sets, which generates a significant number of result points for further analysis.
- the **analysis stage** - containing a number of scripts parsing the acquired results and outputs of the previous stages. The work environment of the analysis is a notebook in Jupyter Notebook which is used both for prototyping scripts and for creating dynamic parameterized graphs.

After executing all stages, the final output are multiple illustrations as well as numerical data summarizing the experiment results. Those are presented as the numerous figures in the following section and Chapter 6.

Both to accelerate the implementation process and to reduce the computation time of consecutive executions after changes in the code, each of the stages can be skipped if its output for a given argument set has already been computed in a previous run. For every execution of the testing system, the stored results are used until one of the stages is entered with new arguments. From that point on, all the results need to be recomputed to account for the change in the implementation.

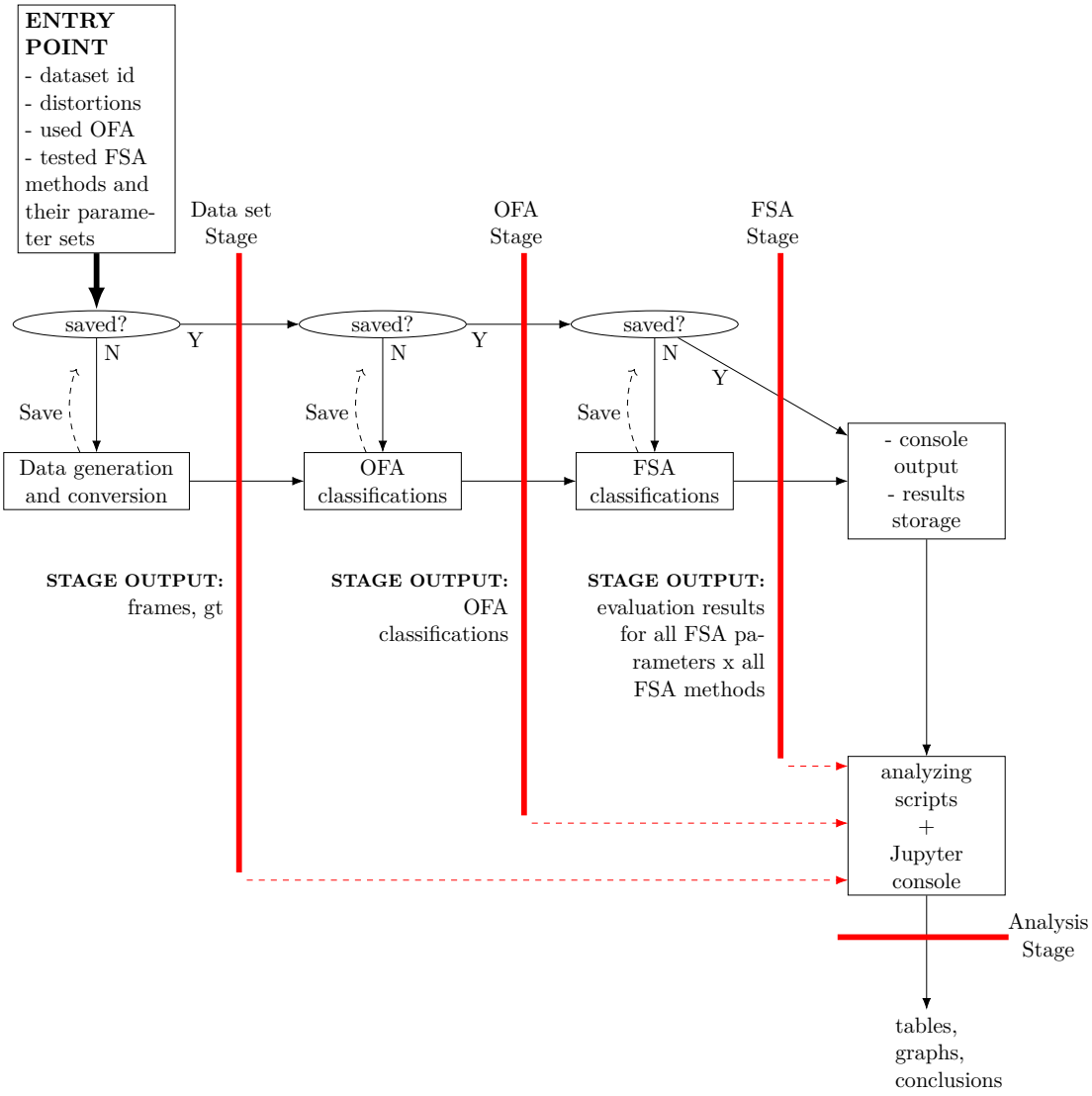


Figure 5.11 – Diagram of the FSA training and evaluation system.

5.4 Preliminary experiments

Due to the broad range of possible algorithms and their parameter sets, the most promising candidates have to be chosen in a faster preliminary experiment. In this section we will present the results of this selection. The goal of it is to establish the most promising and relevant component functions and parameters, which provide the base for further FSA reasoning. Those are:

- the window width range, which keeps the content of the window relevant to the frame of interest;
- a selection of metrics whose values align with the perceived visual continuity of films.

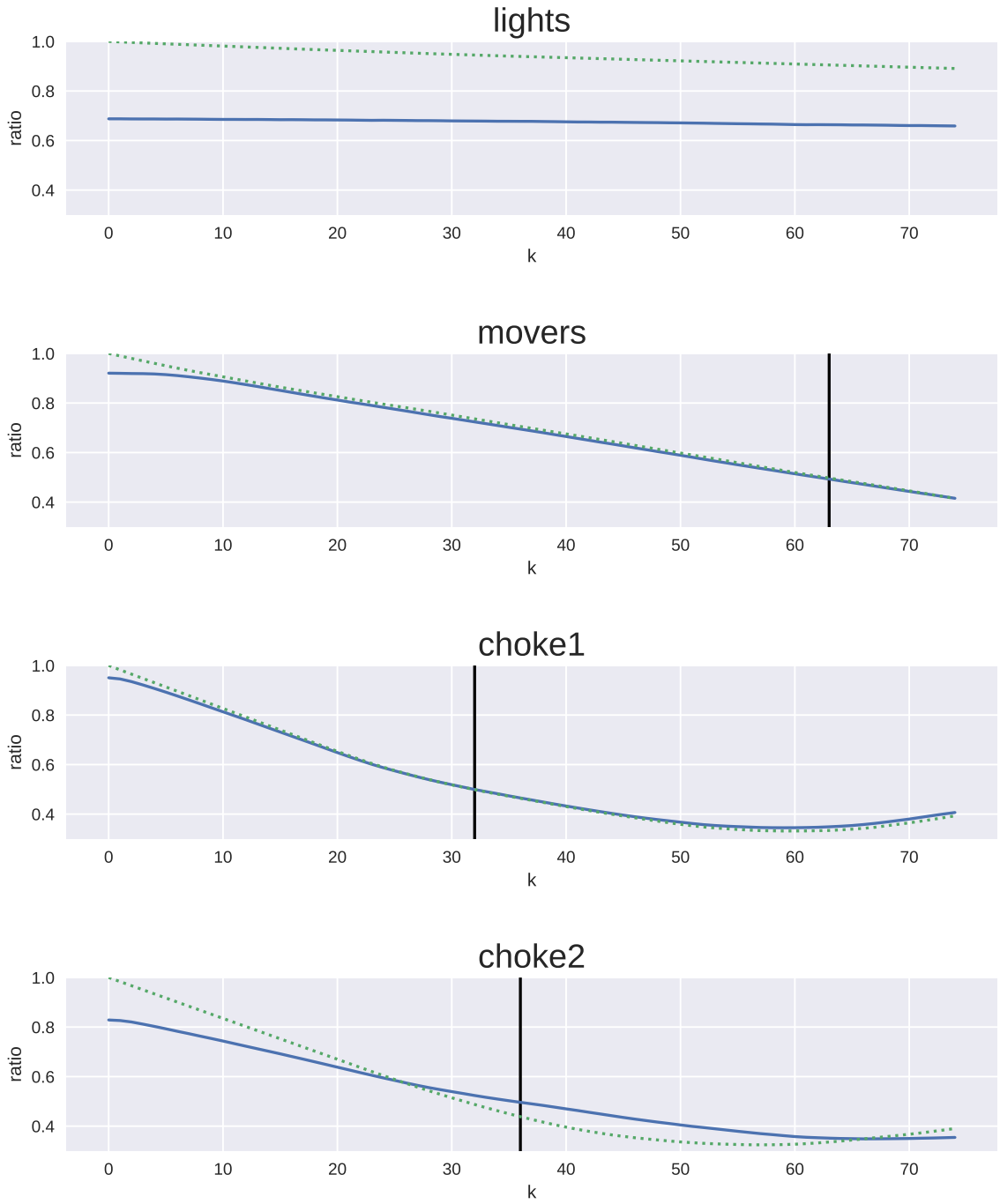


Figure 5.12 – Ratio of classifications matching GT in given distance

5.4.1 Window width range

Given a number of exemplary sequences with provided ground truth classifications, we proceed as follows:

1. For every distance k between frames the ratio is established at which one frame's classifi-

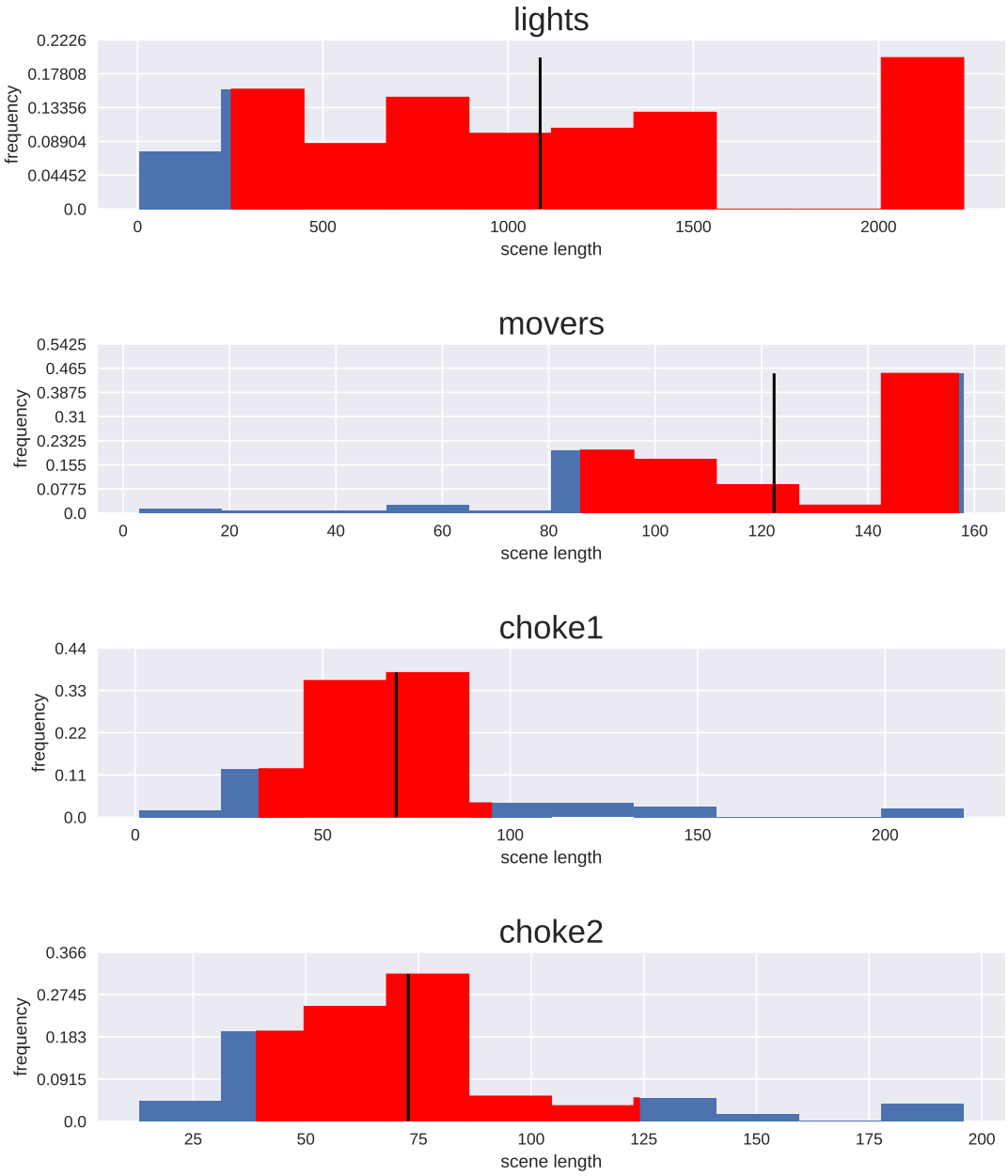


Figure 5.13 – Scene length distributions per data set

cation matches the other frame's ground truth – the results for k in the range $0 \ldots 75$ have been presented in Figure 5.12 (corresponding to $w = 2k + 1 = 1 \ldots 151$). They can be compared with the amount of matches in the corresponding ground truth values (dotted lines in graphs).

2. The upper bound of the cutoff value for the analysis is set on the first value of k, which introduces more confusions than supporting information (ratio of matching frames $< 50\%$). In the case of the TLR (traffic lights) dataset the scenes are much longer, therefore the cutoff value is established at the end of the range (both due to efficiency reasons and to

prevent eliminating shorter scenes).

3. The chosen cutoff value is a value above which the increase of the window size is not contributing to the final FSA classification result anymore - in the case of the analyzed data the majority of windows already extends into scenes with different GT values. Later on some of the graphs climb back up again, but this effect is observed only due to the windows extending into next scenes.

Furthermore, in Figure 5.13 the distribution of scene lengths (as per frame members) of the four input data types has been presented. The vertical black lines on each histogram represent the mean value. The red part of the histogram are the values between the 10th and 90th percentiles.

It is worth noting that the results on both figures are very similar for the two Chokepoint datasets. This confirms that the discussed observations are of a dataset specific nature and not limited to single recordings.

5.4.2 Metric selection

Besides establishing the range of window widths we are interested in, we want to select metric functions which provide a quantitative evaluation of our understanding of image similarity. Such functions are expected to express the similarity between images in each segment of the video and further adjust the priority of votes introduced e.g. by the temporal distance or algorithm confidence.

The distributions of similarity values in neighborhoods of frames have been computed for the following metrics:

- histogram (10 bins for each channel) difference,
- histogram (4 bins for each channel) difference, and
- processed image (blur, downscale, blur again).

Due to performance reasons, multiple similarity functions have had to be eliminated. Only metrics which reduce the image to small vector (that is then kept for further comparisons) prevent the metric computation from strongly dominating the fFSA evaluation time. The chosen represent two different approaches of comparing image similarity (histograms are statistics and the processed image comparison corresponds to a more straightforward difference) and have therefore been chosen as efficient an representative examples.

The means and positional statistics of similarity distances between pairs of frames at given distances from each other have been presented in Figure 5.14. They represent the distribution of similarity k frames away from the central one in a shifting window. It is important to notice that regardless of the different underlying metric functions, the graphs in each case look similar. Again, both Chokepoint datasets show similar results, which indicates that the chosen metrics can return consistent characteristics for a given domain.

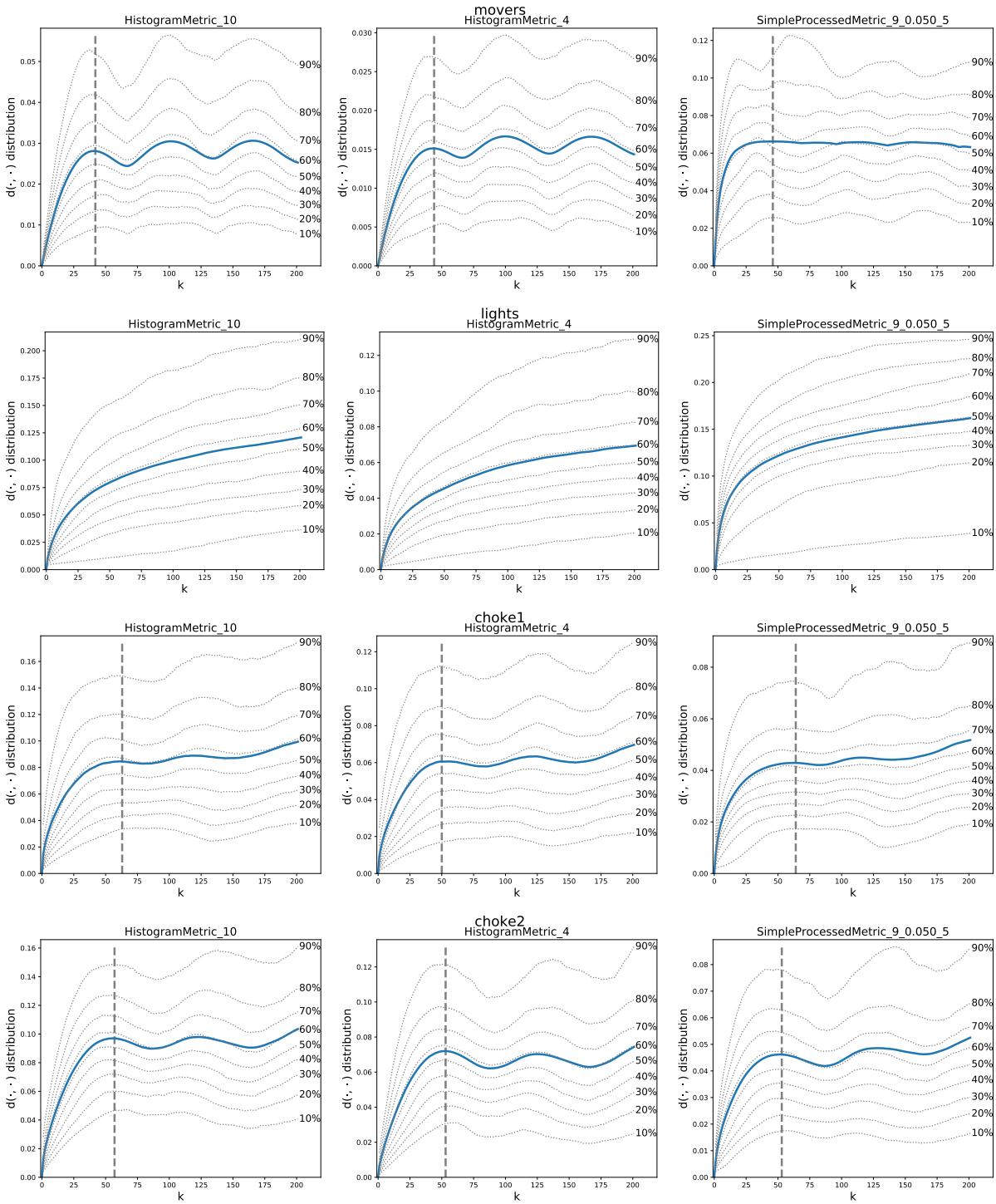


Figure 5.14 – Similarity metric distribution for window

The dashed vertical lines have been placed at the points, where the average value of similarity starts varying (does not anymore steadily increase with k). This value can be considered another upper bound on the window size.

5.5 Tested decision functions

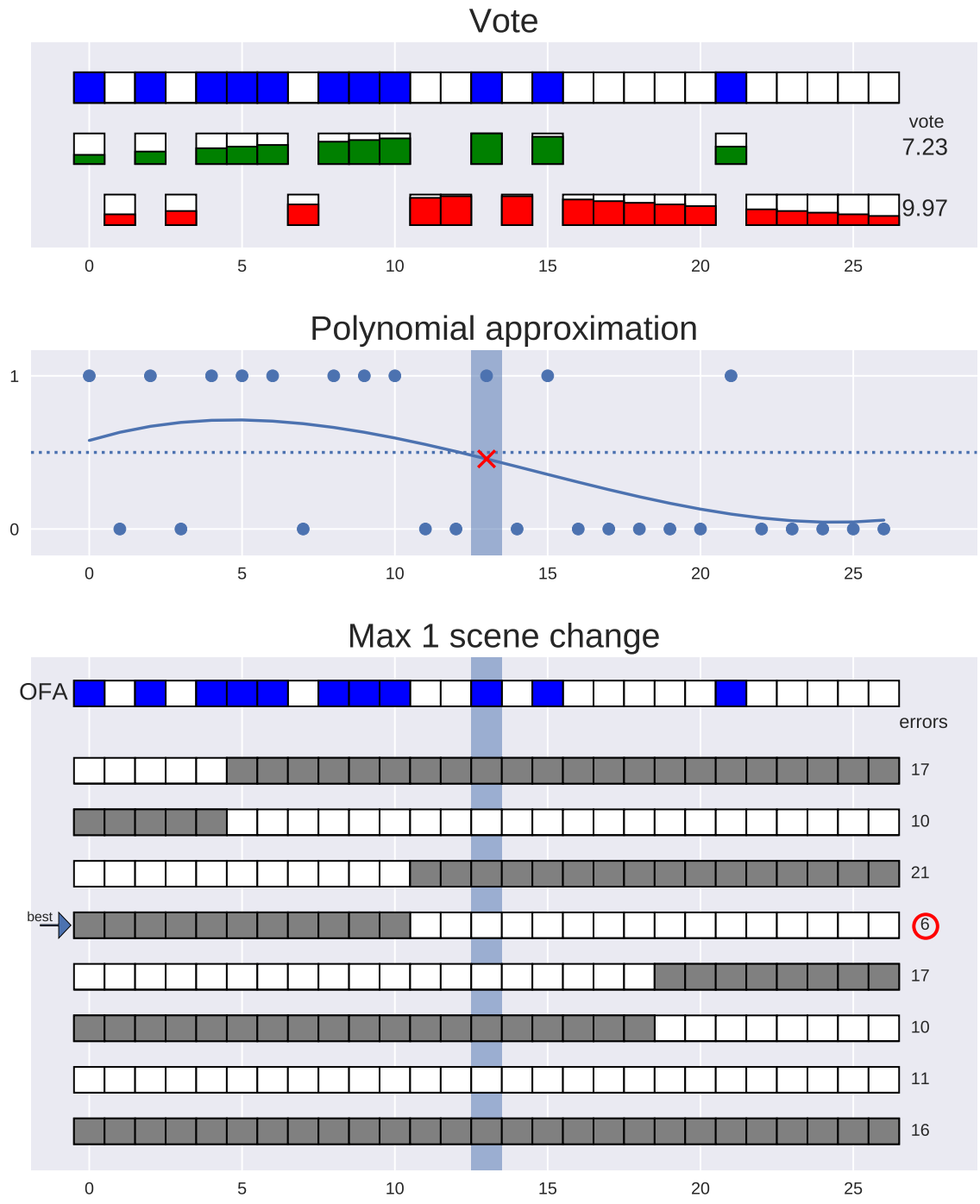


Figure 5.15 – Considered decision functions

Three representative decision functions from the categories proposed in subsection 4.3.4 have been chosen for the evaluation. Figure 5.15 illustrates how each of them analyzes an exemplary input window of 27 frames. The rest of this subsection contains their detailed descriptions.

Weighted vote (vote)

The voting scheme is as proposed in [12] and Algorithm 2.

In the case of iFSA the weights of frames in the windows are defined by a function of their distance from the central frame. The significance distribution D_λ in the presented algorithm has been introduced to represent the decreasing confidence of frames further away from the window's center. For a given distribution parameter $\lambda \in [0; 1]$ it can be expressed as:

$$D_\lambda^m(i) = D_\lambda(i) = 1 - (1 - \lambda) \cdot \frac{|i|}{k} \text{ for } i \in \{-k, \ldots, k\}$$

(i.e. the consecutive weights of a window of width $w = 2k + 1$ are $[\lambda, \ldots, 1 - \frac{(1-\lambda)}{k}, 1, \ldots, \lambda]$, scaling linearly from λ on the ends of the window to 1 in its center).

For fFSA algorithms the distribution is defined by the chosen similarity metric and, again, the parameter $\lambda \in [0; 1]$. The more similar a frame is to the central one, the bigger its weight:

$$D_\lambda^m(i) = \lambda + (1 - \lambda)(1 - \sqrt{d(p_m, p_{m+i})}) \text{ for } i \in \{-k, \ldots, k\}. \tag{5.1}$$

Regardless of the similarity measures value, frames in the shifting window can be assumed to be mostly of similar classifications due to the close temporal proximity. The λ parameter defines the balance between those two factors. It controls the scale of the influence of the similarity on the vote result. The bigger it is, the more equal the influence of all frames on the classification result, regardless of their similarity to the central frame. The square root in the formula has been introduced to emphasize changes for small values of metrics. As it can be seen in Figure 5.14, the normalized metrics rarely reach their maximums.

The weighted vote result for frame m is equal to:

$$\text{vote}(m) = \frac{\sum_{j=-k}^{k} D_\lambda^m(j) O_{m+j}}{\sum_{j=-k}^{k} D_\lambda^m(j)}$$

(m is the index of the central frame in a shifting window, therefore: $k \leq m < n - k$).

The result of the vote is then returned as the result of the decision function. The final decision is established in the next step of the FSA scheme, when the acceptance threshold is applied.

It is worth noting that for $A = 0.5$ and $\lambda = 1$ (and $d(\cdot, \cdot) \equiv 1$) in the proposed FSA method we acquire the majority voting variant discussed in section 3.3.

Numerical approximation (approx)

The sequence of values in the frame is approximated with the function:

$$H(i) = ai^3 + bi^2 + ci + d \tag{5.2}$$

using the value points $i = -k, \ldots, k$ and a least squares polynomial fit. Later on, the central frame's classification gets changed to the rounded value of $H(0)$. The used polynomial is of the third degree to properly model the scope of possible changes within a window (as discussed in Chapter 3) and not overfit.

The value of $H(0)$ before rounding can be interpreted as a measure of certainty - for binary classifications the value of H is closer to 0.5 if there are multiple different classifications in a section of a window or next to scene boundaries. The latter corresponds to the aforementioned ambiguity in the frames around the boundary (it is not always possible to clearly define when the classification of an appearing static feature should change from 0 to 1).

The output of this decion function is thresholded to fit into the $[0; 1]$ range, using the following formula:

$$\text{result} = \max\{0, \min\{1, H(0)\}\}.$$

The final decision is established in the next step of the FSA scheme, when the acceptance threshold is applied.

Model-based scene boundary restriction (model)

Earlier on it has been assumed that a single window almost always contains no more than one scene boundary. Assuming that this condition is kept, we can find a predicted scene boundary for the current window (may be outside of the window) and check which value of the central frame minimizes the number of misclassifications.

The decision result can be acquired by iterating over all possible positions of the single scene boundary (out of the window or between any two frames of the window) and its type (0-1 or 1-0). In case of multiple boundaries acquiring the same accuracy, preserving the original value of the central frame is preferred.

An exemplary evaluation has been presented in Figure 5.15. The OFA classifications (blue) are evaluated in terms of all possible scene boundary positions. The 1-0 scene boundary set after the 11th frame provides the lowest number of discrepancies with the input values. In that scenario the central frame (at the highlighted position) is assigned the value 0, which is the output of the decision function.

This decision function returns only a 0 or a 1, therefore only an artificial acceptance threshold is applied afterwards, which does not change the returned value.

Decision function correlations

After defining the three considered decision functions, we can compare their outputs for multiple smaller window sizes to establish that they are in fact providing different results. Table 5.4 contains the numbers of equal outputs for all pairs of metrics for all window sizes, number of windows with equal answers by all classification functions and numbers of windows with the central frame's classification changed from the initial one:

Table 5.4 – Decision function agreement statistics (ratios of total count). Shortcuts in the table header: v.–vote, a.–approx, m.–model.

						changed		
w	count	v.=a.	v.=m.	a.=m.	v.=a.=m.	v.	a.	m.
5	32	0.81	0.75	0.81	0.69	0.31	0.25	0.06
7	128	0.73	0.75	0.86	0.67	0.34	0.23	0.09
9	512	0.74	0.73	0.82	0.65	0.36	0.29	0.12
11	2048	0.73	0.75	0.83	0.65	0.38	0.30	0.15
13	8192	0.74	0.75	0.82	0.65	0.39	0.33	0.17
15	32768	0.73	0.75	0.82	0.65	0.40	0.34	0.19
17	131072	0.73	0.75	0.82	0.65	0.40	0.35	0.20
19	524288	0.63	0.72	0.81	0.58	0.42	0.36	0.23

From this table we can see that the approx (numerical approximation) and model (model-based scene boundary restriction) decision functions provide the most similar results. Nevertheless, in almost 20% of windows their answers differ. Furthermore, the vote (weighted vote) function changes the most classifications and the model method the fewest.

6 Results

In this chapter we provide the results of a number of experiments performed to evaluate the FSA approach. The first section contains a simple illustration of the reliability of the FSA step even in a very basic case. Next, in section 6.2 an exploratory analysis of the parameters of the voting rule in the iFSA approach is presented. It allows to observe relations between the optimal parameter values and error measures. All of the results are analyzed and deeply discussed, providing graphs for the most interesting observations.

Sections 6.3 and 6.4 address the thesis statements, allowing to confirm the claims from section 2.4. Namely, section 6.3 compares the results of FSA algorithms with their underlying OFA algorithms, establishing the rate of reduced errors. Section 6.4 confirms that the FSA approach is robust to video stream distortions, keeping a comparable classification quality for more distorted streams.

Finally, in section 6.5 the performance of the fFSA variant is discussed. While we can confirm that it improves the classification quality, we also note a number of performance issues, which need to be considered when applying this variant.

6.1 Preliminary experimentation

This section describes the results from [17] without any alterations. The tested algorithm was iFSA with a simple majority voting rule and w=5. The considered case was an endoscopic examination recording, where the underlying OFA algorithm was classifying frames into those which contain visible lesions and those which do not.

The trivial scheme

parameter	**description**
decision rule	Majority Vote
time window	$w = 5$

The most trivial FSA algorithm is the iFSA variant based on a small window size and a simple majority voting rule. Such an approach, as the most intuitive, has found its way into many cited experiments, as it was mentioned in Chapter 2. One such example is [17], the results of which have been presented in Table 6.1.

The evaluation has been performed on the classification results of a very imperfect (low accuracy, especially due to very high FNR, what has been indicated in the paper) OFA algorithm.

Table 6.1 – Results of a simple shifting window experiment. Source: [17].

Method	Number of changed classifications	Number of proper changes	Percent of proper changes
Online (35)	549	**498**	90.7%
Online (15)	358	313	87.4%
Offline (trivial FSA)	53	51	**96.2%**

The algorithms were tested on a video recording of 1500 frames, which were classified as positive if they contained visible lesions and negative if only healthy tissue was present in the view.

Regardless of its apparent simplicity and only with a small window size, the iFSA algorithm (referred to as the "Offline approach" in the cited work) acquired a very good result. While being more conservative than other compared ones, it had the biggest ratio of correct classification changes. Namely 51 out of 53 of its changes in preliminary classifications were correct.

The other compared algorithms (Online with a given parameter value) are domain-specific, designed to increase only the ratio of positive classifications. This resulted in a large total number of changes. Most of them were correct only due to the aforementioned high FNR of the chosen OFA algorithm.

The domain-agnostic FSA algorithm has introduced almost exclusively correct changes to the preliminary classifications. Results such as this one provide an indication of the potential of the FSA approach. In the next section we will analyze the influence of parameters in a more configurable iFSA case.

6.2 Exploratory parameter analysis

This section describes the research from [12], which is already after reviews and accepted for publication in the Signal, Image and Video Processing (SIVP) Springer journal. The tested algorithm was iFSA with the voting rule.

The movers dataset was considered in two variants (AM1, AM2), which were randomly distorted. The goal of the distortions was to increase the error rate of the used OFA algorithm rather than to express the relation of distortions with any phenomena. The distortions were therefore not defined in the same terms as those in this dissertation – their adjustment was subjective, in such a way that AM1 was distorted lighter and AM2 stronger.

The other datasets correspond to those in the main experiments (CP1, CP2 - Chokepoint, TLR - lights), in their undistorted versions.

The results presented below slightly differ from those in the journal, because the L^2 norm has been replaced by rank aggregation.

Within this section we will present and discuss a number of experiment results acquired on

training data for the iFSA variant with the voting decision function. The ranges of possible parameter values have been densely covered with test executions and evaluated. The resulting data has undergone an analysis in terms of relations between the parameters and error measures.

First, pairs of measures have been compared with each other for changing sizes of w in a simple majority vote ($\lambda = 1, A = 0.5$). The graphic presented in Figure 6.1 shows the most informative result of those comparisons – the relation between FPR and FNR with the changing value of w. Both measures improve until a range of values of the window size is reached, for which the quality of the FSA algorithm is at its best. For higher values of w the results change chaotically. The comparison of IBR and MBR has shown chaotic changes of the IBR value and an increase of MBR for increasing values of w.

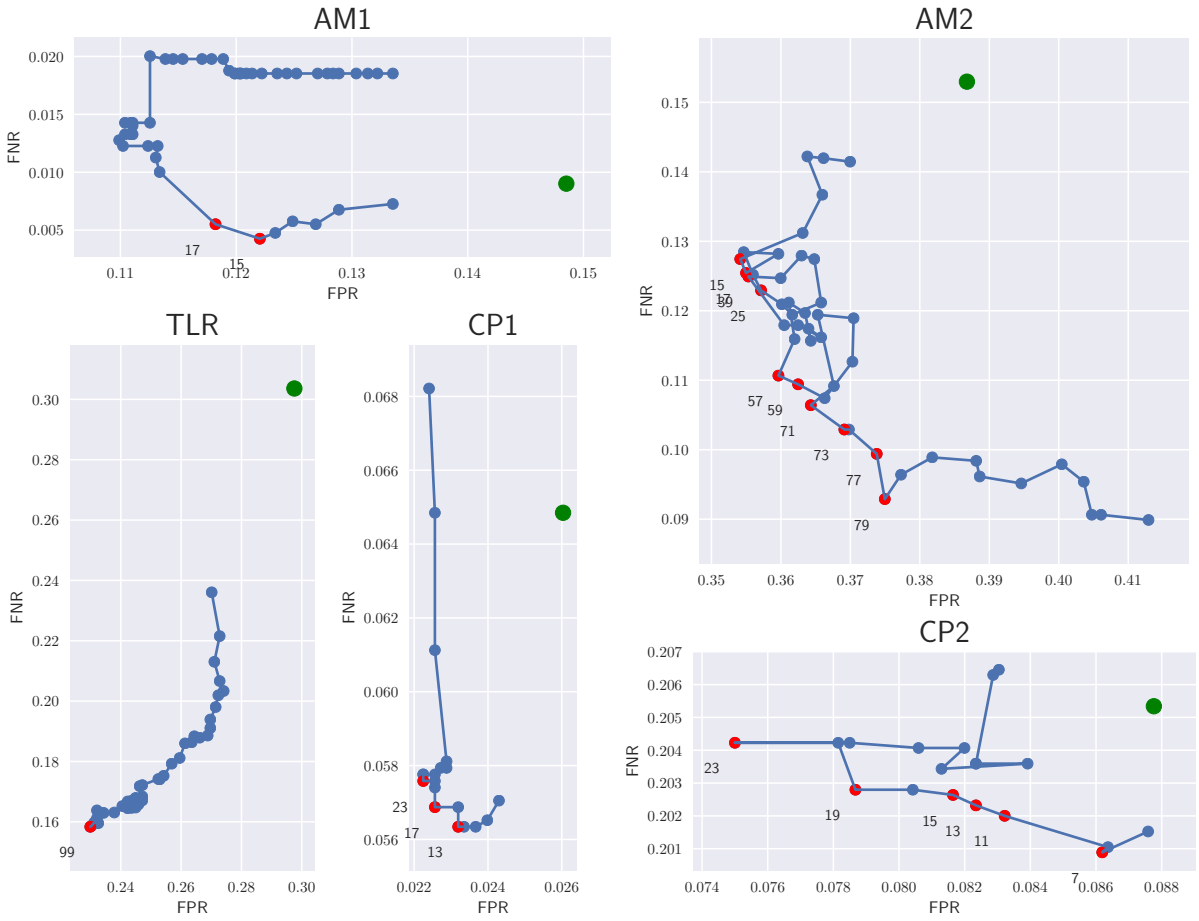


Figure 6.1 – FPR and FNR measures changing with window size (green point - OFA result).

The values shown in Figure 6.1 are two-dimensional, therefore only a partial order can be introduced among them. The red markers represent the minimal results (i.e. there are no other window sizes acquiring a better improvement in terms of both measures at once), which are better than the original OFA. The green points in the graphs represent the corresponding OFA results. The closer the acquired pair of values is to the center of the coordinate system (lower

Table 6.2 – Best window sizes for scene lengths.

Test case	AM1	AM2	TLR	CP1	CP2
Scene lengths	92-154	92-154	469-1425	57-79	53-80
Best window sizes	15-17	15-79	99	13-23	7-23

left side for all graphs), the better the result.

The window sizes which acquired the best results have been compared with the average sizes of scenes in the corresponding recordings in Table 6.2. The scene lengths represent the range from the first to the third quartile. The range of the best values of w correlates with the distribution of the scene size, but is also influenced by the quality of the data.

In Figure 6.2 the best parameter values for each measure and some combinations of measures (All^{rnk} – all; $noMBR^{rnk}$ – all but MBR; *Accuracy* – FPR and FNR) have been presented. The combinations of frame-wise ratios with scene boundaries have been acquired by averaging the ranks of the combined measures.

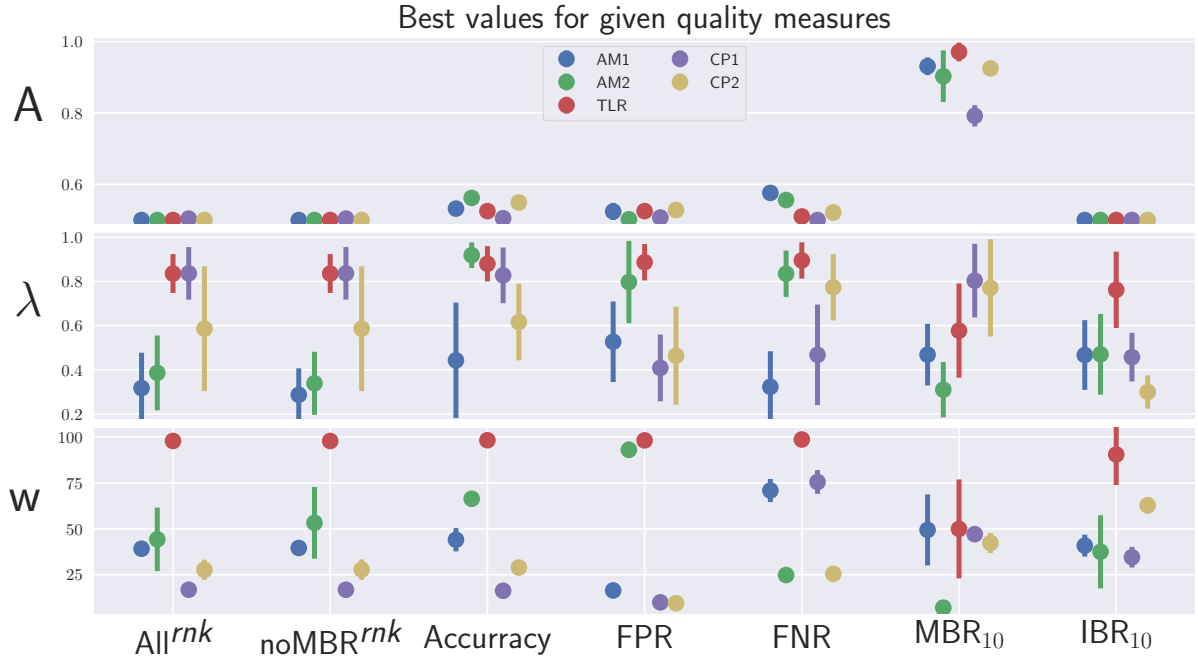


Figure 6.2 – Best parameter values for given measures. Mean and standard deviation range for best 100 results for each case. ($B = 10$)

The results for A show that all measures besides MBR indicate a strong preference for lower values of A. Furthermore, the value of w is the largest for the TLR case for all besides the single MBR measure. Both those observations correspond to the interpretations of the particular measures. The values of λ have the widest ranges and no mean values approach either end of the scale.

Intuitively, the MBR measure was expected to provide a soft limiting effect, so that the FSA step would not smoothen the classification sequence too much. It has been designed symmetrically to IBR, but has been shown to be much more unstable. In most of the experiments its value changed much more than the other ones, dominating the combined measure. The initial OFA classifications have been heavily segmented, therefore boundaries were rarely missing, which resulted in perfect MBR values in the initial data. Due to this strong influence its significance should either be limited or its value provided only as an informative measure after optimizing for the other measures.

To see the underlying relation between the optimal parameters, a plot of the best performing 100 parameter sets for each case has been shown in Figure 6.3. The strong preference for low acceptance threshold values stands out in all cases. Furthermore, in most cases a particular best value of w is visible, which increases slightly with a decreasing value of λ. This relation corresponds to the equivalence of shrinking the window size with decreasing the weights at the ends of the window.

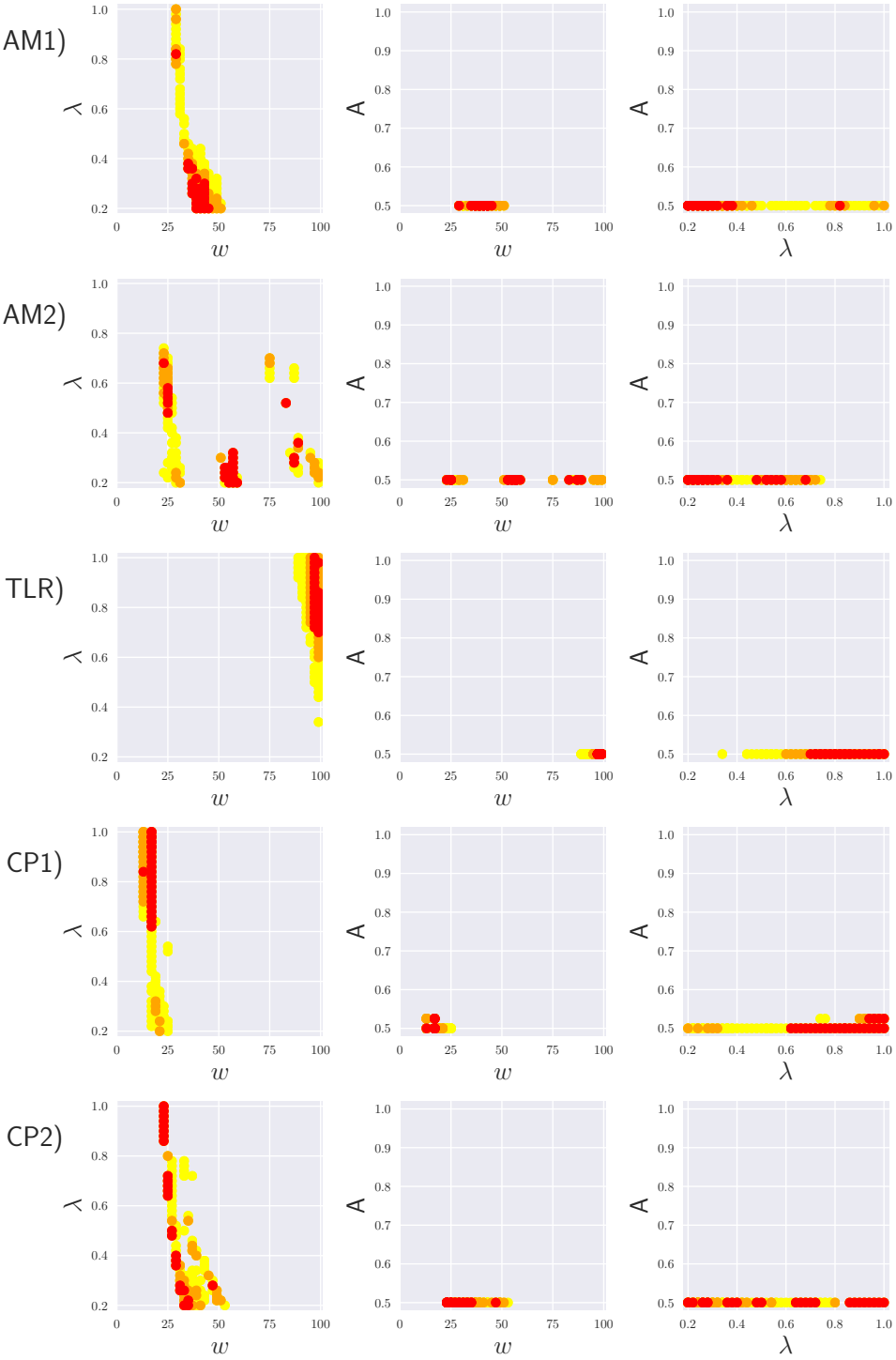


Figure 6.3 – Parameter relations for every dataset ($B = 10$, best reduction of combined FPR, FNR and IBR values expressed by the noMBR^{rank} aggregation; red–best result).

6.3 FSA to OFA comparison (first thesis statement)

In this section we will compare the classification quality of FSA methods with their underlying OFA algorithms. The goal of this section is to confirm the **first thesis statement** – that the FSA step reduces the error measures on average by 20%.

The results of comparing iFSA algorithms with their underlying OFA algorithms have been summarized in Table 6.3. As assumed in the first thesis statement, the FSA approach results in significantly smaller ratios of classification errors. The results of the iFSA decision functions vary and the model-based decision function performs significantly worse than the two other ones[11]. The iFSA approach with the other decision rules never introduced a negative impact.

Table 6.3 – Comparison of optimal iFSA with OFA counterparts – relative values (FSA/OFA).

	method	vote			approx			model		
		FPR	FNR	IBR_B	FPR	FNR	IBR_B	FPR	FNR	IBR_B
B	case									
1	choke1	0.87	0.87	0.57	0.86	0.88	0.57	0.86	0.91	0.72
	choke2	0.98	0.90	0.86	0.98	0.90	0.85	0.99	0.87	0.81
	lights	0.62	0.40	0.97	0.67	0.45	0.99	0.61	0.45	0.99
	movers	0.78	0.89	0.98	0.66	0.91	0.98	0.78	0.91	0.99
	mean	**0.81**	**0.76**	**0.85**	**0.79**	**0.78**	**0.85**	**0.81**	**0.78**	**0.88**
5	choke1	0.87	0.87	0.20	0.86	0.88	0.20	0.86	0.91	0.48
	choke2	0.99	0.88	0.46	0.98	0.90	0.64	0.99	0.88	0.58
	lights	0.62	0.40	0.96	0.62	0.45	0.96	0.61	0.45	0.98
	movers	0.72	0.87	0.72	0.50	0.90	0.71	0.75	0.91	0.80
	mean	**0.80**	**0.76**	**0.58**	**0.74**	**0.78**	**0.63**	**0.80**	**0.78**	**0.71**
10	choke1	0.87	0.87	0.10	0.86	0.88	0.10	0.86	0.91	0.43
	choke2	0.99	0.88	0.39	0.98	0.90	0.58	0.99	0.88	0.51
	lights	0.63	0.39	0.93	0.62	0.44	0.95	0.62	0.45	0.97
	movers	0.69	0.88	0.29	0.50	0.90	0.33	2.12	0.82	0.38
	mean	**0.79**	**0.76**	**0.43**	**0.74**	**0.78**	**0.49**	**1.15**	**0.76**	**0.57**
20	choke1	0.87	0.87	0.04	0.85	0.89	0.04	0.86	0.91	0.40
	choke2	0.99	0.88	0.25	0.98	0.90	0.51	0.99	0.88	0.42
	lights	0.63	0.39	0.90	0.62	0.44	0.91	0.62	0.45	0.96
	movers	0.72	0.87	0.17	0.50	0.90	0.20	1.63	0.84	0.25
	mean	**0.80**	**0.75**	**0.34**	**0.74**	**0.78**	**0.41**	**1.02**	**0.77**	**0.50**
50	choke1	0.87	0.87	0.00	0.85	0.89	0.00	0.86	0.91	0.37
	choke2	0.99	0.88	0.24	0.98	0.90	0.50	1.00	0.88	0.30
	lights	0.63	0.40	0.87	0.62	0.44	0.89	0.61	0.45	0.94
	movers	0.69	0.88	0.13	0.66	0.89	0.13	1.63	0.84	0.20
	mean	**0.79**	**0.76**	**0.31**	**0.78**	**0.78**	**0.38**	**1.03**	**0.77**	**0.45**

For each of the considered boundary tolerances a bold row summarizes the results acquired throughout all cases. This allows to underline the large average impact of the FSA approach on the quality of the results.

Furthermore, and what is most important, the mean results confirm the first assumption from the thesis statements. Although in single cases the error measure values can diverge from the

[11] It is an effect introduced by the aggregation of ranks, which was not observed with the L^2 measure

expected 20% reduction, for the vote and approx decision functions **the FSA approach never worsens the results and on average reduces the error rates by at least about 20%**.

The only exceptions are cases with $B = 1$, which correspond to an extremely strict boundary tolerance. As it was already noted in section 2.1, boundaries are usually ambiguous and therefore those results don't weigh heavily on the conclusion[12].

The results acquired by the iFSA algorithms chosen as optimal have also been shown in absolute units, in Table 6.4. It is worth to note that the worse results of the model-based decision function are a result of not improving the very small FPR rate in the movers case. The rows in bold fonts present averages of the corresponding FSA/OFA results. Computing the ratios would yield a different result than that which can be found in the bold rows in Table 6.3 (ratio of averages vs average of ratios). We can observe that the averages of absolute OFA and FSA results remain very stable throughout the choice of decision functions and change mostly with the value of B. This indicates that the variability of means in Table 6.3 is caused mostly by the improvement ratios of smaller errors.

Table 6.4 – Comparison of optimal iFSA with OFA counterparts – absolute values (FSA/OFA).

	method	vote			approx			model		
		FPR	FNR	IBR_B	FPR	FNR	IBR_B	FPR	FNR	IBR_B
B	case									
1	choke1	0.07/0.08	0.02/0.03	0.36/0.64	0.06/0.08	0.02/0.03	0.36/0.64	0.06/0.08	0.03/0.03	0.46/0.64
	choke2	0.24/0.24	0.09/0.10	0.68/0.79	0.24/0.24	0.09/0.10	0.67/0.79	0.24/0.24	0.08/0.10	0.64/0.79
	lights	0.19/0.31	0.08/0.21	0.97/1.00	0.21/0.31	0.09/0.21	0.99/1.00	0.19/0.31	0.09/0.21	0.98/1.00
	movers	0.01/0.01	0.11/0.13	0.93/0.95	0.01/0.01	0.12/0.13	0.93/0.95	0.01/0.01	0.12/0.13	0.95/0.95
	mean	**0.12/0.16**	**0.08/0.12**	**0.74/0.84**	**0.13/0.16**	**0.08/0.12**	**0.74/0.84**	**0.12/0.16**	**0.08/0.12**	**0.76/0.84**
5	choke1	0.07/0.08	0.02/0.03	0.10/0.51	0.06/0.08	0.02/0.03	0.10/0.51	0.06/0.08	0.03/0.03	0.24/0.51
	choke2	0.24/0.24	0.09/0.10	0.29/0.63	0.24/0.24	0.09/0.10	0.40/0.63	0.24/0.24	0.08/0.10	0.37/0.63
	lights	0.19/0.31	0.08/0.21	0.95/0.99	0.19/0.31	0.09/0.21	0.96/0.99	0.19/0.31	0.09/0.21	0.97/0.99
	movers	0.01/0.01	0.11/0.13	0.60/0.84	0.00/0.01	0.11/0.13	0.60/0.84	0.01/0.01	0.12/0.13	0.67/0.84
	mean	**0.12/0.16**	**0.08/0.12**	**0.49/0.74**	**0.12/0.16**	**0.08/0.12**	**0.52/0.74**	**0.12/0.16**	**0.08/0.12**	**0.56/0.74**
10	choke1	0.07/0.08	0.02/0.03	0.05/0.49	0.06/0.08	0.02/0.03	0.05/0.49	0.06/0.08	0.03/0.03	0.21/0.49
	choke2	0.24/0.24	0.09/0.10	0.23/0.60	0.24/0.24	0.09/0.10	0.35/0.60	0.24/0.24	0.08/0.10	0.31/0.60
	lights	0.19/0.31	0.08/0.21	0.93/0.99	0.19/0.31	0.09/0.21	0.95/0.99	0.19/0.31	0.09/0.21	0.96/0.99
	movers	0.01/0.01	0.11/0.13	0.21/0.72	0.00/0.01	0.11/0.13	0.24/0.72	0.02/0.01	0.10/0.13	0.27/0.72
	mean	**0.13/0.16**	**0.08/0.12**	**0.35/0.70**	**0.12/0.16**	**0.08/0.12**	**0.39/0.70**	**0.13/0.16**	**0.08/0.12**	**0.44/0.70**
20	choke1	0.07/0.08	0.02/0.03	0.02/0.47	0.06/0.08	0.02/0.03	0.02/0.47	0.06/0.08	0.03/0.03	0.19/0.47
	choke2	0.24/0.24	0.09/0.10	0.14/0.57	0.24/0.24	0.09/0.10	0.29/0.57	0.24/0.24	0.08/0.10	0.24/0.57
	lights	0.19/0.31	0.08/0.21	0.89/0.99	0.19/0.31	0.09/0.21	0.90/0.99	0.19/0.31	0.09/0.21	0.95/0.99
	movers	0.01/0.01	0.11/0.13	0.12/0.68	0.00/0.01	0.11/0.13	0.13/0.68	0.01/0.01	0.11/0.13	0.17/0.68
	mean	**0.13/0.16**	**0.08/0.12**	**0.29/0.68**	**0.12/0.16**	**0.08/0.12**	**0.33/0.68**	**0.13/0.16**	**0.08/0.12**	**0.39/0.68**
50	choke1	0.07/0.08	0.02/0.03	0.00/0.46	0.06/0.08	0.02/0.03	0.00/0.46	0.06/0.08	0.03/0.03	0.17/0.46
	choke2	0.24/0.24	0.09/0.10	0.14/0.56	0.24/0.24	0.09/0.10	0.28/0.56	0.24/0.24	0.09/0.10	0.17/0.56
	lights	0.19/0.31	0.08/0.21	0.86/0.99	0.19/0.31	0.09/0.21	0.88/0.99	0.19/0.31	0.09/0.21	0.94/0.99
	movers	0.01/0.01	0.11/0.13	0.09/0.68	0.01/0.01	0.11/0.13	0.09/0.68	0.01/0.01	0.11/0.13	0.14/0.68
	mean	**0.13/0.16**	**0.08/0.12**	**0.27/0.67**	**0.12/0.16**	**0.08/0.12**	**0.31/0.67**	**0.13/0.16**	**0.08/0.12**	**0.35/0.67**

[12] In one of the works related with this dissertation [15], it has also been noted as a conclusion that the frames around scene boundaries introduce unnecessary false classifications, when correct algorithms happen to be adjusted different than the ground truth.

Plots summarizing the relative improvements as markers on the (OFA result, FSA result) planes for every error measure have been presented in Figure 6.4. The results in this figure have been acquired by optimizing the classification for the noMBR$^{\text{rnk}}$ measure set. It is worth to compare them with those in Figure 6.5, which were acquired by optimizing the parameters separately for each of the measures. Treating those results as a reference, it stands out that in most cases the values optimized for noMBR$^{\text{rnk}}$ have already acquired very good results. This indicates that the optimization in terms of the three measures is close to optimal for all of them.

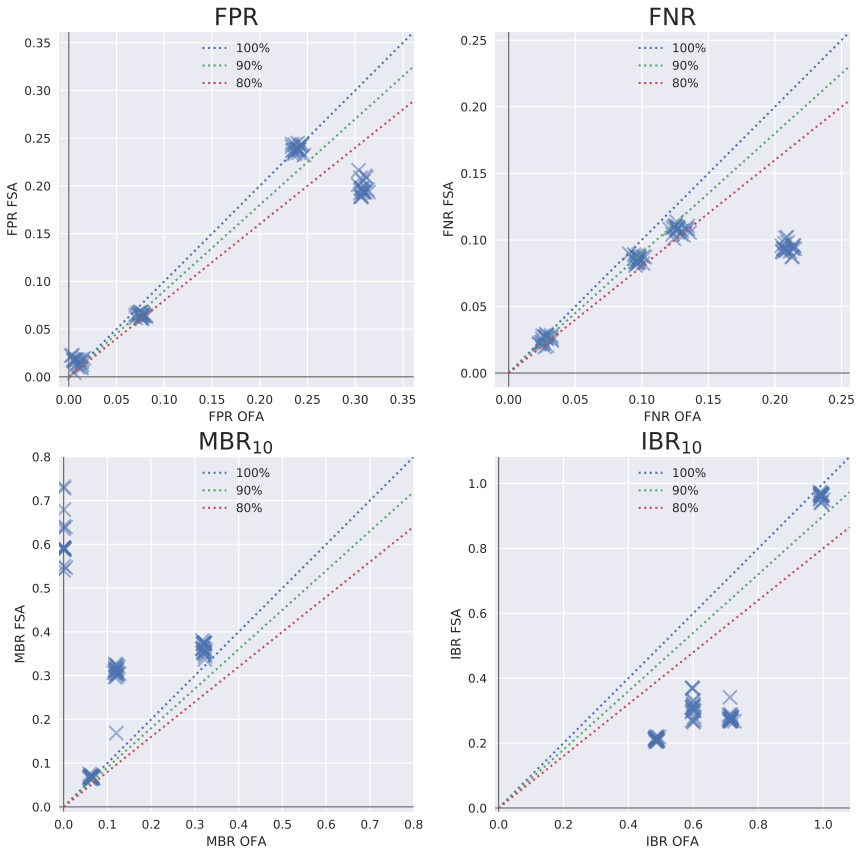


Figure 6.4 – FSA to OFA error measure relations for all datasets (optimized for the noMBR$^{\text{rnk}}$ measure combination).

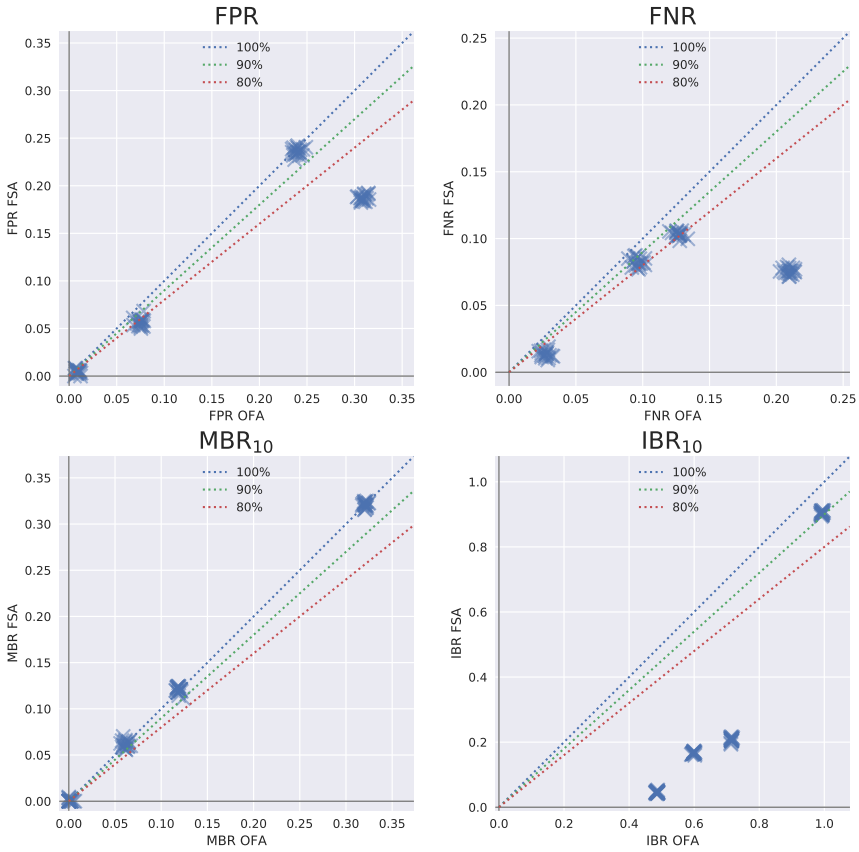


Figure 6.5 – FSA to OFA error measure relations for all datasets (optimized separately for every one of the considered measures).

6.4 Distortion influence (second thesis statement)

In this section we establish the robustness of the FSA methods to the distortions, which were defined in Chapter 5. The goal of this evaluation is to confirm the **second thesis statement** – that the FSA algorithms keep the quality of their underlying OFA algorithms even when the video stream is 10 intensity percentage points more distorted.

Influence on OFA classifications

The classification quality measure values for all datasets have been summarized in Table 6.5, Table 6.6 and Table 6.7. To introduce a consistent reference point, all values have been normalized so that each non-distorted case, that is – (0,0), acquires the value 1.

All of the applied distortions result in a decrease of the overall classification quality. Still, the results presented in the table underline the significance of considering the error measure set as a whole. For example, the FPR measure gets seemingly improved in multiple cases. Only the comparison with the corresponding FNR value explains, that the detection algorithm returns an excessive amount of negative answers on the distorted video. This kind of relation is domain- and OFA-specific, depending on the OFA algorithm's details and the semantics of a positive finding. As discussed in subsection 4.7.4, the changes introduced by distortions can amount both to an actual change in classification quality, as well as a shift on the ROC curve. Therefore, the measure values in the tables have been complemented with value of the RMS, which combines the FPR and FNR measures.

The results for the IBR measure did not change as significant for any of the distortions. This is a result of the imperfect OFA algorithms, which return an overly segmented classification sequences even for videos which are not distorted.

The MBR values for the lights case have been given only in terms of absolute values, because there are no missing scene boundaries in the undistorted stream. Therefore, the normalized ratio couldn't be established.

It has to be noted that acquiring and classifying the distorted videos with an OFA algorithm is the most expensive and time-consuming part of the computations. As it can be seen in Figure 5.11, executing tests with additional distortions takes us back to the very beginning of the testing procedure. Therefore, even though the distortions are stochastic processes whose evaluation would benefit from repeated runs, the number of their executions is limited. Similar trends can be observed in other fields requiring extensive computations, which contain a random component and require days of computation. For example, the performance of deep neural networks is usually reported as a singular number, even though the initialization of weights is based on a random seed.

In the case of evaluating the FSA approach we can also note, that the provided error measures are statistics of stochastic processes. Consecutive realizations (e.g. true/false positive/negative) are dependent variables, but at temporal distances larger than typical scene lengths individual

Table 6.5 – Influence of quality distortions on OFA results. Distortion levels (x,0).

		Normalized to (0,0)					Absolute				
	x (%)	0	10	20	30	40	0	10	20	30	40
	case										
FPR	choke1	1.00	0.99	0.46	0.39	0.38	0.08	0.07	0.03	0.03	0.03
	choke2	1.00	0.37	0.38	0.28	0.25	0.24	0.09	0.09	0.07	0.06
	lights	1.00	0.72	0.90	1.37	1.46	0.31	0.22	0.28	0.42	0.45
	movers	1.00	6.53	3.41	2.38	1.78	0.01	0.05	0.03	0.02	0.01
FNR	choke1	1.00	3.56	11.13	16.65	20.67	0.03	0.10	0.31	0.46	0.58
	choke2	1.00	2.75	4.75	6.14	6.82	0.10	0.27	0.46	0.59	0.66
	lights	1.00	2.15	2.74	1.99	1.88	0.21	0.45	0.57	0.42	0.39
	movers	1.00	1.07	1.31	1.54	1.97	0.13	0.14	0.17	0.20	0.25
RMS	choke1	1.00	1.55	3.89	5.80	7.19	0.06	0.09	0.22	0.33	0.41
	choke2	1.00	1.08	1.80	2.30	2.55	0.18	0.20	0.33	0.42	0.47
	lights	1.00	1.35	1.71	1.59	1.60	0.26	0.36	0.45	0.42	0.42
	movers	1.00	1.15	1.33	1.54	1.97	0.09	0.10	0.12	0.14	0.18
IBR	choke1	1.00	1.46	1.47	1.55	1.51	0.49	0.71	0.71	0.75	0.73
	choke2	1.00	1.28	1.36	1.32	1.29	0.60	0.77	0.81	0.79	0.77
	lights	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	0.99	0.99
	movers	1.00	1.21	1.18	1.22	1.27	0.72	0.86	0.85	0.87	0.91
MBR	choke1	1.00	1.45	3.27	5.09	5.91	0.06	0.09	0.20	0.32	0.37
	choke2	1.00	0.53	1.09	1.55	1.94	0.32	0.17	0.35	0.50	0.62
	lights	-	-	-	-	-	0.00	0.05	0.32	0.27	0.27
	movers	1.00	0.42	0.92	1.50	1.58	0.12	0.05	0.11	0.18	0.19

Table 6.6 – Influence of obscuring distortions on OFA results. Distortion levels (0,y).

		Normalized to (0,0)					Absolute				
	y (%)	0	10	20	30	40	0	10	20	30	40
	case										
FPR	choke1	1.00	1.06	0.99	0.87	0.77	0.08	0.08	0.07	0.07	0.06
	choke2	1.00	0.95	0.84	0.80	0.62	0.24	0.23	0.20	0.19	0.15
	lights	1.00	1.04	1.10	1.15	1.19	0.31	0.32	0.34	0.36	0.37
	movers	1.00	6.47	19.28	26.59	27.97	0.01	0.05	0.15	0.21	0.22
FNR	choke1	1.00	4.36	9.42	12.53	15.42	0.03	0.12	0.26	0.35	0.43
	choke2	1.00	1.67	2.54	3.41	4.48	0.10	0.16	0.24	0.33	0.43
	lights	1.00	1.04	1.03	1.18	1.13	0.21	0.22	0.21	0.25	0.24
	movers	1.00	1.10	1.16	1.28	1.34	0.13	0.14	0.15	0.16	0.17
RMS	choke1	1.00	1.81	3.40	4.43	5.41	0.06	0.10	0.19	0.25	0.31
	choke2	1.00	1.08	1.23	1.47	1.76	0.18	0.20	0.22	0.27	0.32
	lights	1.00	1.04	1.07	1.16	1.17	0.26	0.27	0.28	0.31	0.31
	movers	1.00	1.18	1.67	2.11	2.21	0.09	0.11	0.15	0.19	0.20
IBR	choke1	1.00	1.58	1.75	1.78	1.79	0.49	0.77	0.85	0.87	0.87
	choke2	1.00	1.38	1.48	1.50	1.49	0.60	0.83	0.89	0.90	0.89
	lights	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99
	movers	1.00	1.23	1.29	1.31	1.32	0.72	0.88	0.92	0.93	0.94
MBR	choke1	1.00	0.91	0.73	1.27	1.91	0.06	0.06	0.05	0.08	0.12
	choke2	1.00	0.62	0.26	0.47	0.58	0.32	0.20	0.08	0.15	0.19
	lights	-	-	-	-	-	0.00	0.00	0.00	0.05	0.00
	movers	1.00	0.75	0.58	0.67	0.42	0.12	0.09	0.07	0.08	0.05

Table 6.7 – Influence of combined distortions on OFA results. Distortion levels (x,x).

		Normalized to (0,0)						Absolute					
	x (%)	0	10	20	30	40	50	0	10	20	30	40	50
	case												
FPR	choke1	1.00	0.80	0.59	0.43	0.37	0.42	0.08	0.06	0.04	0.03	0.03	0.03
	choke2	1.00	0.46	0.29	0.27	0.30	0.22	0.24	0.11	0.07	0.07	0.07	0.05
	lights	1.00	0.89	0.98	1.32	1.42	1.20	0.31	0.27	0.30	0.41	0.44	0.37
	movers	1.00	16.12	13.94	15.81	10.28	9.72	0.01	0.13	0.11	0.13	0.08	0.08
FNR	choke1	1.00	8.27	17.74	22.59	26.19	27.77	0.03	0.23	0.49	0.63	0.73	0.77
	choke2	1.00	3.65	5.96	6.89	7.51	7.94	0.10	0.35	0.57	0.66	0.72	0.77
	lights	1.00	1.74	2.50	2.11	2.00	2.19	0.21	0.37	0.52	0.44	0.42	0.46
	movers	1.00	1.18	2.18	2.11	3.36	3.28	0.13	0.15	0.28	0.27	0.43	0.42
RMS	choke1	1.00	2.97	6.19	7.86	9.11	9.66	0.06	0.17	0.35	0.44	0.52	0.55
	choke2	1.00	1.43	2.23	2.58	2.81	2.97	0.18	0.26	0.41	0.47	0.51	0.54
	lights	1.00	1.22	1.63	1.61	1.63	1.58	0.26	0.32	0.43	0.43	0.43	0.42
	movers	1.00	1.55	2.35	2.33	3.41	3.33	0.09	0.14	0.21	0.21	0.31	0.30
IBR	choke1	1.00	1.72	1.78	1.80	1.79	1.80	0.49	0.83	0.86	0.87	0.87	0.87
	choke2	1.00	1.43	1.48	1.49	1.51	1.49	0.60	0.86	0.89	0.89	0.90	0.89
	lights	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99
	movers	1.00	1.29	1.33	1.33	1.34	1.33	0.72	0.92	0.95	0.95	0.96	0.95
MBR	choke1	1.00	1.55	2.91	5.18	6.27	8.00	0.06	0.10	0.18	0.32	0.39	0.50
	choke2	1.00	0.45	1.04	1.40	1.68	1.74	0.32	0.15	0.33	0.45	0.54	0.56
	lights	-	-	-	-	-	-	0.00	0.05	0.05	0.23	0.36	0.27
	movers	1.00	0.42	0.50	0.58	1.58	1.50	0.12	0.05	0.06	0.07	0.19	0.18

classifications can be considered independent. This observation indicates that extensions of the central limit theorem could be considered for supporting the provided measures. Therefore the reported results can be treated as more informative than a single draw from a random distribution.

Influence on FSA classifications

The next results allow us to address the second thesis statement. We expect that the FSA algorithm is robust to distortions in such a way, that it performs as well as its underlying OFA even when the video for the FSA evaluation is distorted more by 10 intensity percent points. This result will be presented in terms of both quality and obscuring distortions. Furthermore, also the performance of FSA on streams distorted stronger (in terms of both kinds of distortions at once) will be evaluated.

The collected measurements of average error rates for different distortion intensities have been presented in Figure 6.6. The presented graphs compare the averaged OFA results with the averaged optimal FSA results. FSA results marked with green circles corresponds to cases, where the FSA algorithm performed at least just as good as OFA on the stream less distorted by the corresponding 10 percent points. The red crosses mark those results where this is not the case. Those results are presented in terms of the FPR, FNR and IBR measures. The majority of comparison falls in favor of the FSA algorithm on the more distorted stream.

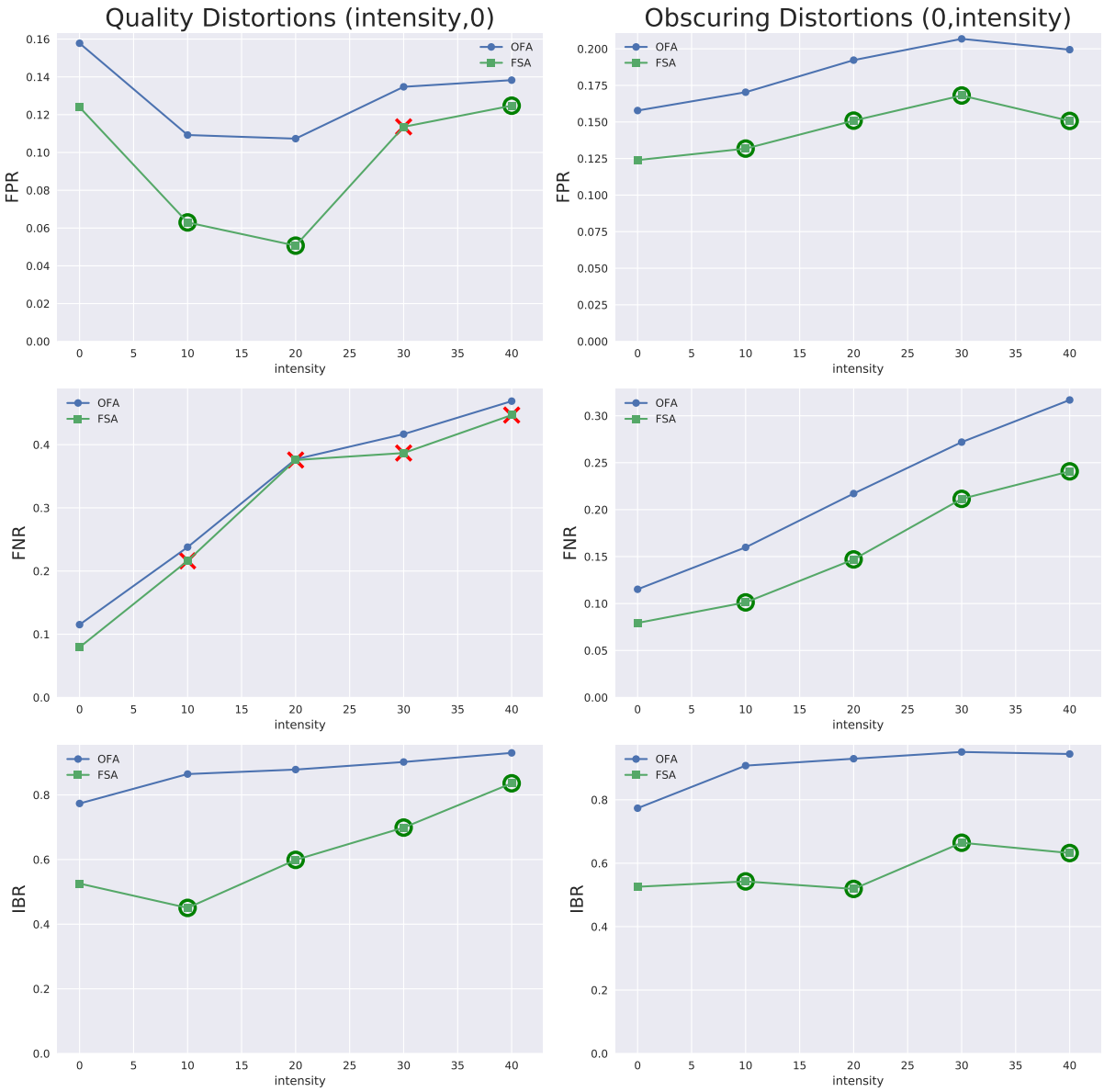


Figure 6.6 – Changes in classification quality for different distortion intensities.

A similar comparison has been performed separately for every considered case. The individual plots are grouped in Figure 6.7. Also in this case the majority of comparisons favor the FSA result. Nevertheless, the FNR results on both Chokepoint recordings need to be pointed out, where the OFA approach on the less distorted stream is consequently favored. Those cases correspond with the results presented in Table 6.3 and Table 6.9, which also show that in the Chokepoint cases the FSA approach introduces little improvement in terms of the FNR error.

The results of all the comparisons are summarized in Table 6.8. To confirm the second thesis statement at least a tie between the FSA on the more distorted and OFA on the less distorted stream is required. As it can be seen, this has been acquired, what **confirms the second thesis**

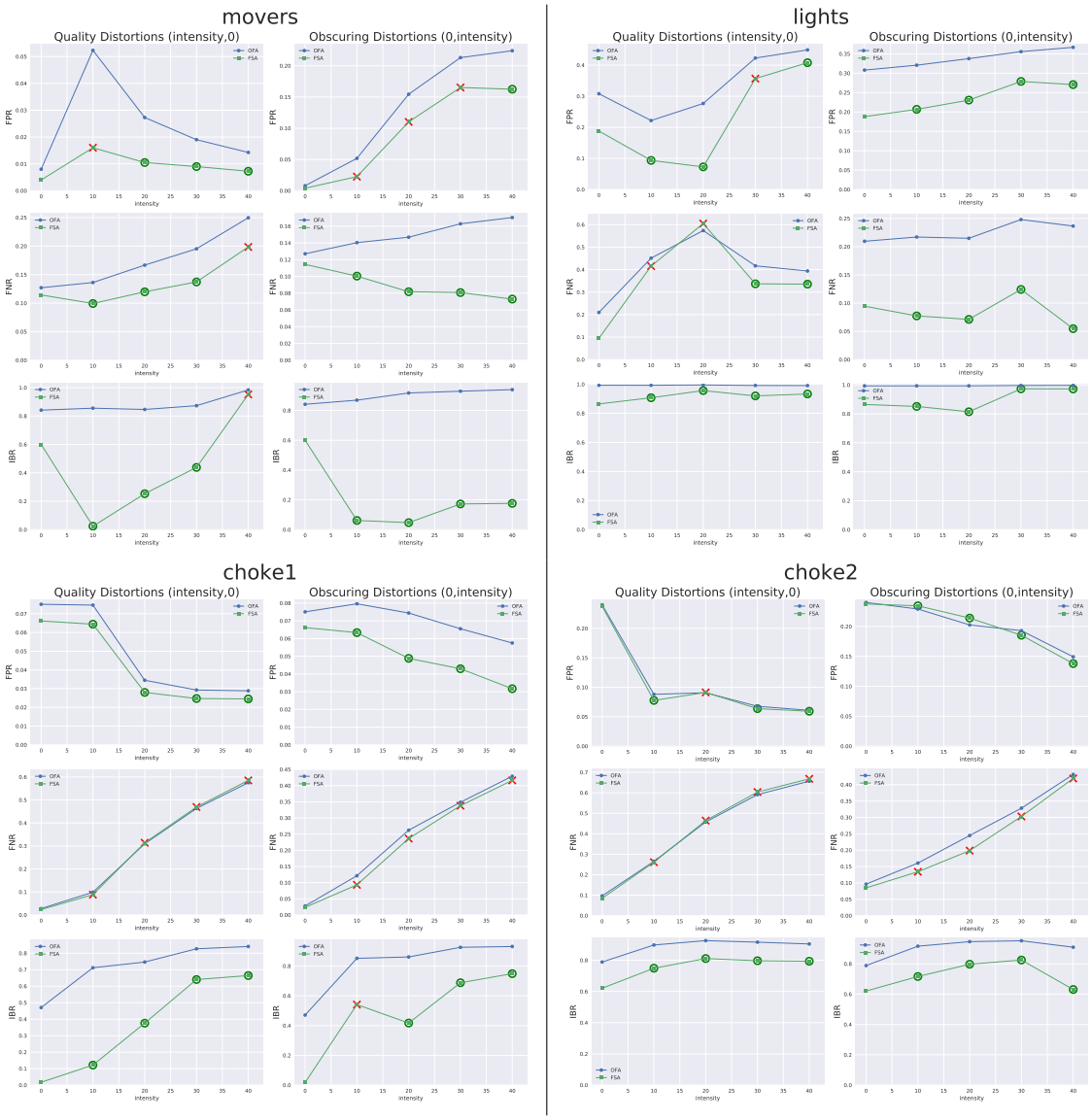


Figure 6.7 – Changes in classification quality for different distortion intensities - per case.

statement. The same experiment has been repeated with both intensity values being changed simultaneously. Its results have been presented with same type of graph in Figure 6.8 (only one column for each case and measure, because the intensity value is the same in both dimensions). The combined effects of the distortions alter the quality of the OFA and FSA algorithms in a manner, which corresponds to the sum of the effects of two distortion types separately. Also in this case, the majority of comparisons confirms the second thesis statement (although 10 percent points are added to two the intensities distortions at once), with the Chokepoint cases providing the least support.

We will now continue with a brief extended discussion of the results.

In Figure 6.6 and Figure 6.7, we can observe that not all of the relations are monotonous.

Table 6.8 – Number of measurements, where FSA retains performance on the more distorted stream.

case	measure FNR	FPR	IBR	ratio
joined	**4**	**7**	**8**	0.79
choke1	0	8	7	0.62
choke2	0	7	8	0.62
lights	6	7	8	0.88
movers	7	4	7	0.75

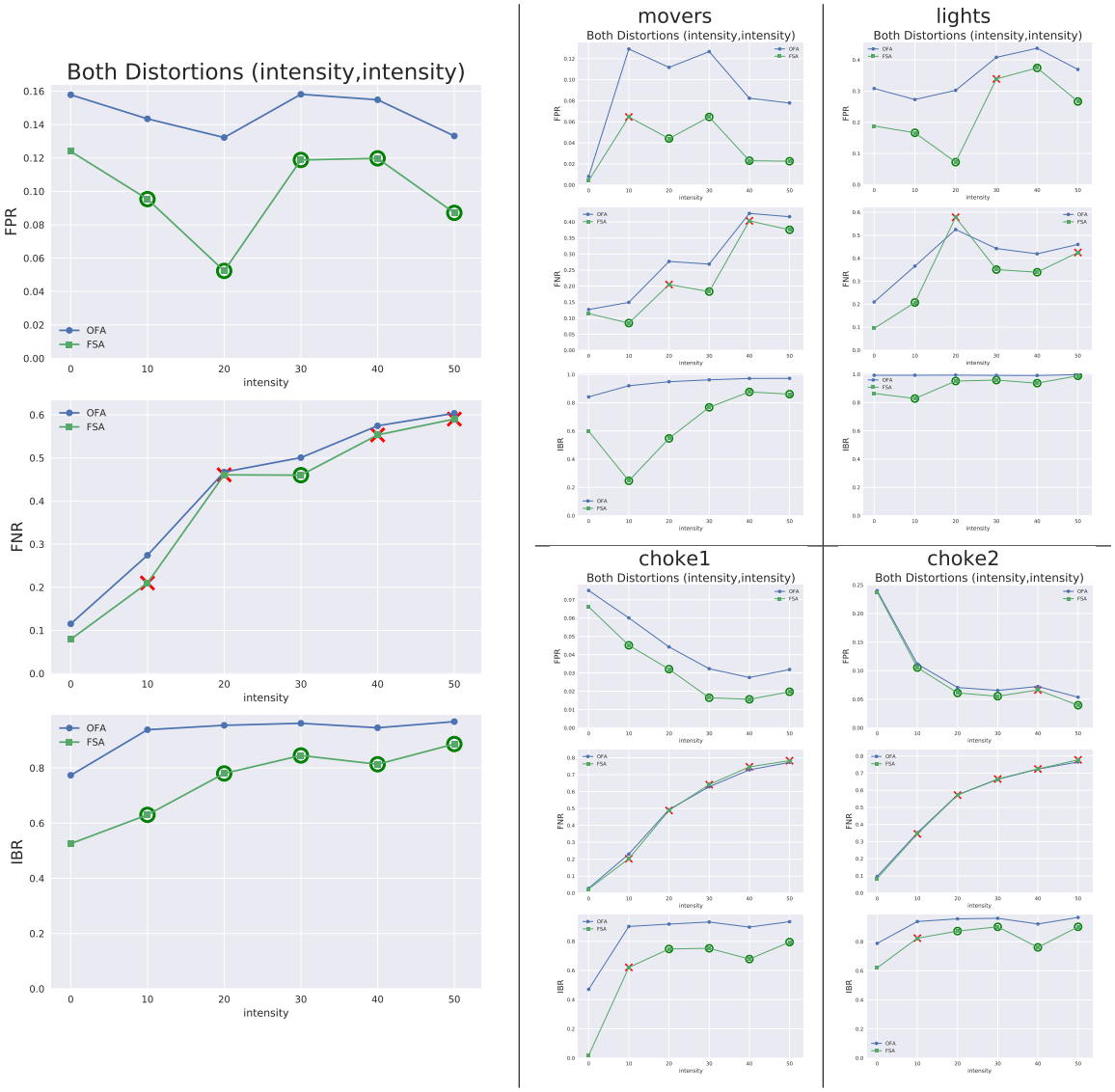


Figure 6.8 – Changes in classification quality for a steady increase of both distortion intensities.

This can be mostly explained by the inner workings of the OFA classifiers and the multiple effects distortions have on them.

When defining the OFA classifiers, a soft convention was followed which assigns the positive class to the presence of particular features. That means that a plain image (filled only with one color) would rather be classified as negative. We can propose a simple distinction of impairing influences, which an classifying algorithm can undergo:

- random increase of positive classifications,
- random increase of negative classifications,
- randomized classifications.

Among the considered types of distortions, some remove information from the image (e.g. by obscuring important areas with artifacts, blurring edges) and others add new elements (e.g. the edges of artifacts are very distinctive, salt-and-pepper pixels have a high contrast). We can say very generally that the former ones reduce and the latter ones increase the number of detections/positive answers of the classifier. The overlap of those effects shapes the graphs for each case in Figure 6.7 and Figure 6.8. For instance, the TLR detector in the lights case first experiences a drop in the FPR and increase in the FNR values, because the spot lights get distorted and loose their saliency. After the quality distortion intensity is increased further, spots with more bright "salt" pixels and an increasingly blurred background (which reduces the local variances, what influences the adaptive template matching) become new detections. This in turn results in the increase of the FPR value for higher distortion intensity values.

Another observation concerns the robustness of the FSA method to distortions. The graphs of the error measures acquired by FSA are in most cases closely following the shape of the corresponding OFA results. As described in subsection 5.2.5, the distortions in the experiments cover virtually all frames in the distorted videos. In this case, the observed robustness of the FSA approach comes from it improving the OFA algorithm at a rate larger than that of how distortions worsen its results. With such a density of distortions the FSA approach is still away from its theoretical capabilities, which are approximated by the evaluation in Figure 3.8.

The results of this evaluation would have been even better, if (in some cases more realistic) short distortions were applied. Even if every 2 out of 5 consecutive frames were to acquire wrong classifications, error rates of up to 40% could be fully reduced. Such an effect is a trivial consequence of the FSA definition.

6.5 fFSA results

The fFSA methods are a superset of the iFSA methods, therefore the best results of the former class are always at least as good as those of the latter. A comparison of the fFSA-specific results with their corresponding OFA results has been presented in Table 6.9 in the same manner as we have compared the OFA and iFSA methods above. We can observe that also in this case the first thesis statement has been fulfilled for all cases except $B = 1$, where some error levels remain slightly higher than desired.

Table 6.9 – Comparison of optimal fFSA with OFA counterparts

	method	fFSA(HD_{10})			fFSA(HD_4)			fFSA(SP)		
		FPR	FNR	IBR_B	FPR	FNR	IBR_B	FPR	FNR	IBR_B
B	case									
1	choke1	0.86	0.88	0.58	0.86	0.88	0.58	0.86	0.89	0.57
	choke2	0.98	0.90	0.87	0.99	0.89	0.78	0.99	0.89	0.80
	lights	0.62	0.40	0.97	0.62	0.40	0.97	0.62	0.40	0.97
	movers	0.78	0.89	0.98	0.56	0.90	0.98	0.78	0.89	0.98
	mean	**0.81**	**0.77**	**0.85**	**0.76**	**0.77**	**0.83**	**0.81**	**0.77**	**0.83**
5	choke1	0.86	0.88	0.21	0.86	0.88	0.21	0.86	0.89	0.19
	choke2	0.99	0.89	0.49	0.99	0.89	0.47	0.99	0.89	0.50
	lights	0.62	0.40	0.96	0.62	0.40	0.96	0.62	0.40	0.96
	movers	0.59	0.89	0.72	0.56	0.89	0.72	0.69	0.88	0.71
	mean	**0.76**	**0.76**	**0.59**	**0.76**	**0.76**	**0.59**	**0.79**	**0.76**	**0.59**
10	choke1	0.87	0.87	0.10	0.87	0.87	0.10	0.86	0.89	0.10
	choke2	0.99	0.89	0.42	0.99	0.89	0.40	0.99	0.89	0.43
	lights	0.62	0.40	0.95	0.62	0.40	0.95	0.62	0.40	0.95
	movers	0.69	0.88	0.29	0.69	0.88	0.29	0.69	0.88	0.29
	mean	**0.79**	**0.76**	**0.44**	**0.79**	**0.76**	**0.43**	**0.79**	**0.76**	**0.44**
20	choke1	0.87	0.87	0.04	0.87	0.87	0.04	0.86	0.89	0.04
	choke2	0.99	0.85	0.34	0.99	0.89	0.28	0.99	0.89	0.32
	lights	0.62	0.40	0.93	0.62	0.40	0.93	0.62	0.40	0.92
	movers	0.59	0.89	0.17	0.56	0.89	0.17	0.69	0.88	0.14
	mean	**0.77**	**0.75**	**0.37**	**0.76**	**0.76**	**0.35**	**0.79**	**0.76**	**0.36**
50	choke1	0.87	0.87	0.00	0.87	0.87	0.00	0.86	0.89	0.00
	choke2	0.99	0.85	0.33	0.99	0.89	0.26	0.99	0.89	0.31
	lights	0.62	0.40	0.88	0.62	0.40	0.88	0.62	0.40	0.87
	movers	0.69	0.88	0.13	0.69	0.88	0.13	0.69	0.88	0.13
	mean	**0.79**	**0.75**	**0.33**	**0.79**	**0.76**	**0.32**	**0.79**	**0.76**	**0.33**

In this table, the first outstanding (and somewhat surprising) result is that the three decision functions acquired virtually identical results. The Simple Processed metric is significantly different from the Histogram Difference metric – they were therefore expected to emphasize different kinds of similarity within the shifting windows. As it turns out, regardless of the measure the results hardly differ. In most cases they are very close to the results acquired by the iFSA variant with the voting decision rule[13]. This seems to indicate that the voting component of the fFSA rule (the part adjusted with the λ parameter, first coefficient in Equation 5.1) is the

[13]The results acquired with the L^2 norm have been even closer, but this observation still holds.

most influential for the classification quality. Table 6.10 contains the absolute values of the corresponding evaluations.

Table 6.10 – Comparison of optimal fFSA with OFA counterparts – absolute values (FSA/OFA)

	method	fFSA(HD_{10})			fFSA(HD_4)			fFSA(SP)		
		FPR	FNR	IBR_B	FPR	FNR	IBR_B	FPR	FNR	IBR_B
B	case									
1	choke1	0.06/0.08	0.02/0.03	0.37/0.64	0.06/0.08	0.02/0.03	0.37/0.64	0.06/0.08	0.02/0.03	0.36/0.64
	choke2	0.24/0.24	0.09/0.10	0.69/0.79	0.24/0.24	0.09/0.10	0.62/0.79	0.24/0.24	0.09/0.10	0.63/0.79
	lights	0.19/0.31	0.08/0.21	0.97/1.00	0.19/0.31	0.08/0.21	0.97/1.00	0.19/0.31	0.08/0.21	0.97/1.00
	movers	0.01/0.01	0.11/0.13	0.93/0.95	0.00/0.01	0.11/0.13	0.94/0.95	0.01/0.01	0.11/0.13	0.94/0.95
	mean	**0.12/0.16**	**0.08/0.12**	**0.74/0.84**	**0.12/0.16**	**0.08/0.12**	**0.72/0.84**	**0.12/0.16**	**0.08/0.12**	**0.72/0.84**
5	choke1	0.06/0.08	0.02/0.03	0.11/0.51	0.06/0.08	0.02/0.03	0.11/0.51	0.06/0.08	0.02/0.03	0.10/0.51
	choke2	0.24/0.24	0.09/0.10	0.31/0.63	0.24/0.24	0.09/0.10	0.30/0.63	0.24/0.24	0.09/0.10	0.32/0.63
	lights	0.19/0.31	0.08/0.21	0.95/0.99	0.19/0.31	0.08/0.21	0.95/0.99	0.19/0.31	0.08/0.21	0.95/0.99
	movers	0.00/0.01	0.11/0.13	0.60/0.84	0.00/0.01	0.11/0.13	0.60/0.84	0.01/0.01	0.11/0.13	0.59/0.84
	mean	**0.12/0.16**	**0.08/0.12**	**0.49/0.74**	**0.12/0.16**	**0.08/0.12**	**0.49/0.74**	**0.12/0.16**	**0.08/0.12**	**0.49/0.74**
10	choke1	0.07/0.08	0.02/0.03	0.05/0.49	0.07/0.08	0.02/0.03	0.05/0.49	0.06/0.08	0.02/0.03	0.05/0.49
	choke2	0.24/0.24	0.09/0.10	0.25/0.60	0.24/0.24	0.09/0.10	0.24/0.60	0.24/0.24	0.09/0.10	0.26/0.60
	lights	0.19/0.31	0.08/0.21	0.94/0.99	0.19/0.31	0.08/0.21	0.94/0.99	0.19/0.31	0.08/0.21	0.94/0.99
	movers	0.01/0.01	0.11/0.13	0.21/0.72	0.01/0.01	0.11/0.13	0.21/0.72	0.01/0.01	0.11/0.13	0.21/0.72
	mean	**0.12/0.16**	**0.08/0.12**	**0.36/0.70**	**0.12/0.16**	**0.08/0.12**	**0.36/0.70**	**0.12/0.16**	**0.08/0.12**	**0.36/0.70**
20	choke1	0.07/0.08	0.02/0.03	0.02/0.47	0.07/0.08	0.02/0.03	0.02/0.47	0.06/0.08	0.02/0.03	0.02/0.47
	choke2	0.24/0.24	0.08/0.10	0.19/0.57	0.24/0.24	0.09/0.10	0.16/0.57	0.24/0.24	0.09/0.10	0.18/0.57
	lights	0.19/0.31	0.08/0.21	0.92/0.99	0.19/0.31	0.08/0.21	0.92/0.99	0.19/0.31	0.08/0.21	0.92/0.99
	movers	0.00/0.01	0.11/0.13	0.12/0.68	0.00/0.01	0.11/0.13	0.12/0.68	0.01/0.01	0.11/0.13	0.10/0.68
	mean	**0.12/0.16**	**0.08/0.12**	**0.31/0.68**	**0.12/0.16**	**0.08/0.12**	**0.30/0.68**	**0.12/0.16**	**0.08/0.12**	**0.30/0.68**
50	choke1	0.07/0.08	0.02/0.03	0.00/0.46	0.07/0.08	0.02/0.03	0.00/0.46	0.06/0.08	0.02/0.03	0.00/0.46
	choke2	0.24/0.24	0.08/0.10	0.18/0.56	0.24/0.24	0.09/0.10	0.15/0.56	0.24/0.24	0.09/0.10	0.17/0.56
	lights	0.19/0.31	0.08/0.21	0.87/0.99	0.19/0.31	0.08/0.21	0.87/0.99	0.19/0.31	0.08/0.21	0.87/0.99
	movers	0.01/0.01	0.11/0.13	0.09/0.68	0.01/0.01	0.11/0.13	0.09/0.68	0.01/0.01	0.11/0.13	0.09/0.68
	mean	**0.12/0.16**	**0.08/0.12**	**0.29/0.67**	**0.12/0.16**	**0.08/0.12**	**0.28/0.67**	**0.12/0.16**	**0.08/0.12**	**0.28/0.67**

The execution of the FSA step for the fFSA variant requires to perform $O(wn)$ similarity evaluations. This has limited the range of considered similarity metrics for this evaluation. The currently tested metrics all start with a step which dramatically reduces the dimensionality of the comparison – from a whole frame to a fraction of the frame's size. More complex measures require more computing power, more time for processing, or an approximation approach. The last option can be implemented by evaluating the similarity of all frames in a video with a small fixed set of reference frames. Exploiting the triangle inequality can allow to acquire an estimate of the actual distance between two given pictures.

There are no significant differences in the classification quality, and the computing time of the fFSA method is significantly longer (up to 10 times difference between the iFSA and fFSA variants on the post-processing step). Therefore the proposed fFSA method with the tested similarity metrics does not add additional value when compared to the simpler iFSA with the voting rule.

While the results for the fFSA method confirm the thesis statements, we did not acquire a

strong indication of a noteworthy improvement when compared to the iFSA variant. Nevertheless, the definition of the method allows for replacing the component functions freely, which could lead to further improvements in future work. This opens the question about new similarity measures, which would be both precise and efficient.

Proposing the fFSA variant is a result of looking for further fields of improvement after the successful evaluations of iFSA. Therefore, the discussions and experiments within this dissertation are aimed mostly at proving its potential and providing grounds for further research in this topic. The wide overview of applicable image similarity metrics in Chapter 4 shows the wealth of available approaches and encourages to further investigate in this field.

Furthermore, another research question can be formulated regarding the approach to the similarities within a frame. Currently, the weights are adjusted according to the similarity to the central frame. If it is heavily distorted, it can be significantly different than those correctly classified. Therefore, a new approach can be proposed, which uses the inter-frame similarities to cluster frames within a window, and use the biggest/best scored cluster as an indication regarding the proper classification of the central frame. This observation has been noted in the future work perspectives in the next chapter.

6.6 Additional analysis of the FSA approach

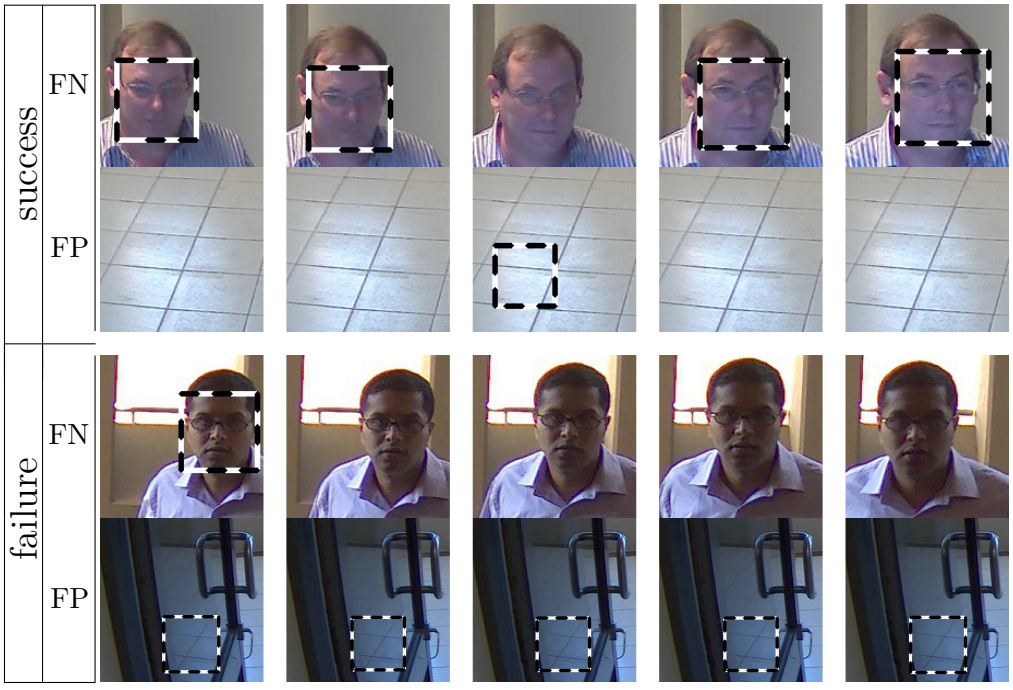


Figure 6.9 – Example OFA outputs: FSA success and failure cases. Source: [12]

Figure 6.9 shows exemplary cases from the choke2 dataset, where the FSA approach succeeds

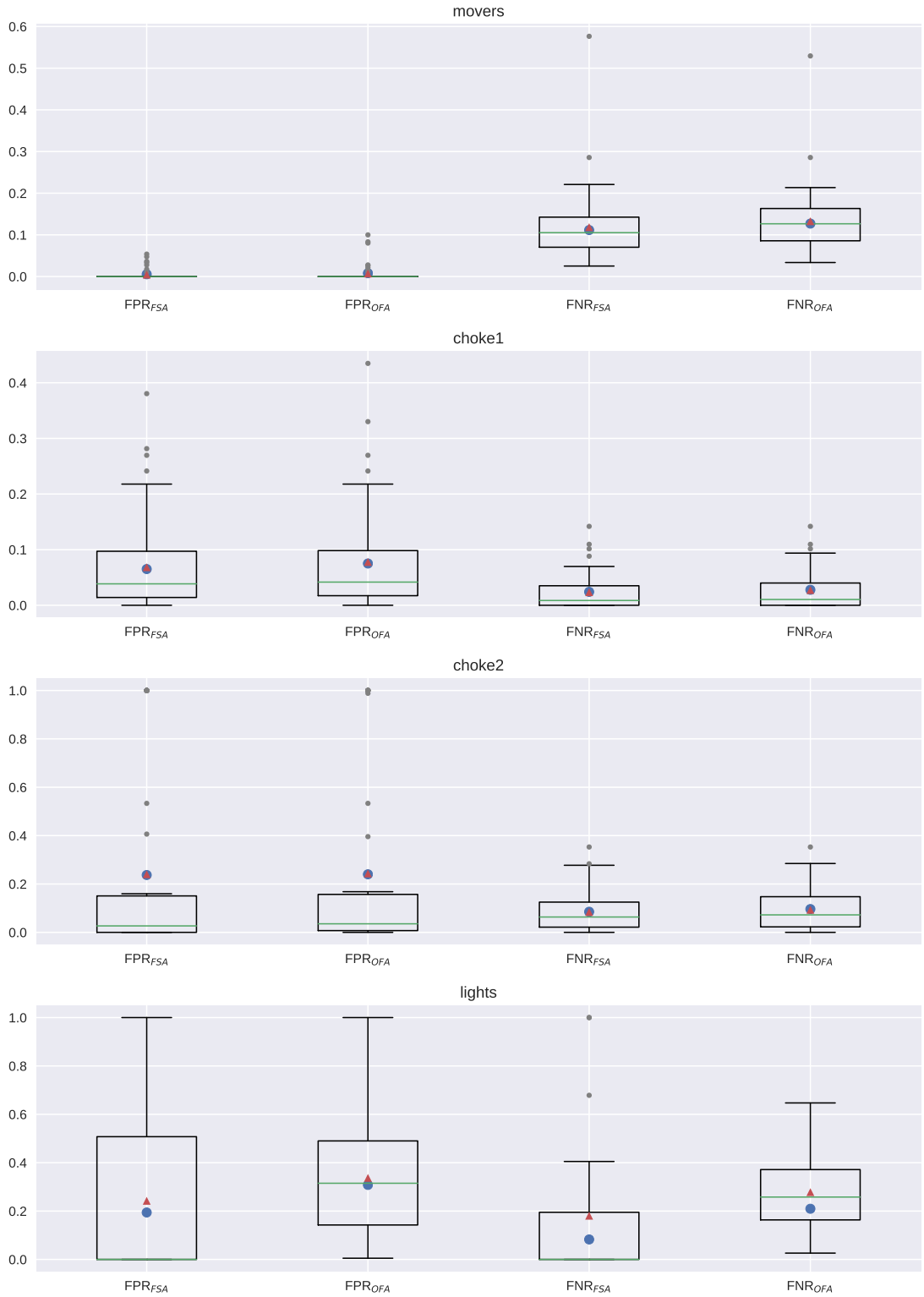


Figure 6.10 – Distributions of FSA and OFA results in sequences of 200 frames.

and fails in correcting classifications. The black and white frame in each frame fragment indicates the (possibly wrong) face detection, which caused a positive OFA classification. We can notice that the OFA detector makes short mistakes due to the variability of the appearance of detected objects in real-life videos–which is also why it is susceptible to the FSA improvement. Nevertheless, the FSA step is not able to correct long-term systemic errors. Especially when the OFA algorithm fixates on generating false detections in a semi-static background (as in the failure/FP example), the amount of generated misclassifications becomes significant. In cases such as the failure/FN example a single face detection in the sequence will be treated as the outlier and the central frame of the window will keep the false negative classification.

Figure 6.10 presents statistics of the classification results of the best FSA and corresponding OFA algorithms in terms of sequences of 200 frames. The boxes represent the ranges of values from the 25th to the 75th percentile, the green lines – the medians. The grey outlier points are identified as those out of ranges proportional to the boxes' sizes (away from the box limits by more than 1.5 times the box's span). The red points are mean values of the measurements, the blue ones – results acquired by the given algorithm on the whole recording.

Overall, the classification improvements acquired in terms of the global score can also be observed in the distributions. The value ranges and positional statistics shift towards 0, what corresponds to a uniform improvement. The outlier points correspond to the aforementioned long sequences of misclassifications. As stated, we can observe that in many cases they are barely altered by the FSA step. Especially in the case of choke2/FPR, when the face detector gets stuck on wrongfully identifying a part of the background as a positive detection, full subsequences are getting false positive ratios of 100% which cannot be corrected. Those outliers significantly influence the total score, which ends up above the majority of local results. This leads to the conclusion, that the FSA approach is the most suitable for OFA algorithms with a relatively high accuracy and low stability of results. This ensures the best possible gain in the course of the FSA improvement.

6.7 Comparison with other approaches

Thus far, no general methods for improving OFA algorithms have been proposed which could be compared with the FSA approach without a domain-specific context. Results in different works are comparable only to a limited degree, as evaluations are performed on different datasets, underlying algorithms and with domain-specific measures. Nevertheless, single works can be cited which explicitly evaluate the influence of improving classifications by utilizing temporal properties of the video – which is the essence of the FSA method.

The temporal filtering of Bourennane and Fossati [22] improved the hand gesture recognition rate from 84.7% to 87.5% (which would correspond to reducing the FNR by 18%, without a known change in FPR or result sequence segmentation).

The simple median temporal filter in [101] improved the overlapping area ratio of object

detections by 4.31% and a global HMM method by 9.64%. These measures are not directly translatable to any of those used in this dissertation[14].

In [119] a post-processing step ensures that scenes have a length of at least 100 frames. It is stated that this kind of outlier post-processing "slightly lowers" the FNR (alas, again without information regarding other quality measures).

In a different field (pedestrian reidentification [47]), but with an approach related to ours, an increase of the F-score values from 15.5% to 19.2% and from 25.6-28.1% to 33.5-38.9% has been acquired. The quality parameters of the improved algorithms were much lower than those of the OFA algorithms used in this work. This is due to the much more complex problem (assigning identities to detected bounding boxes with silhouettes).

The results acquired by the FSA approach are similar to those presented above. This shows that the FSA step correctly and universally (as it is not bound to a particular domain) utilizes the temporal relations in the video.

It has to be noted that all of the OFA algorithms which have been improved are either de facto industry standards or state-of-the-art representatives – the face and silhouette detectors are widely used and the TLR algorithm has thus far not been outperformed by other proposed methods. The fact that the FSA approach was able to improve all of them is a strong confirmation of the proposed method.

Exact comparisons are hindered by the lack of a standard testing set. Furthermore, methods other than FSA require much bigger amounts of training data, what increases the cost of preparing such a comparison. This issue at the same time underlines the advantage of the FSA approach, which can be optimized already with small amounts of ground truth data.

[14] We can only loosely relate the individual pixel assignments to object/non-object classes in this work with binary classifications, and conclude that the resulting score shows an improvement of a similar order of magnitude as that of the FSA approach.

7 Conclusions

Accomplishments

Within this dissertation a new, original method for improving classifications of images in video streams has been proposed. By pointing out multiple works proposing the usage of OFA algorithms, we have identified a possibility of improving their results with a universal approach. It has been named Frame Sequence Analysis (FSA), as it considers the frame classification sequence in terms of its temporal structure and the continuous real-life process it represents.

The analysis of existing works and research indicates that the FSA approach is an original contribution to the subject of video frame classification. It has been identified, defined and analyzed for the first time. Therefore, besides implementing a number of representative FSA algorithms, we have provided a broad discussion of possible implementations and extensions.

The performed experiments have proven the thesis statements, regarding both the quality improvement introduced by FSA, as well as the proposed method's robustness to video distortions. For this, a complex testing environment has been designed and implemented, which allows for an efficient optimization and evaluation of the FSA approach.

The FSA method

The FSA method is characterized by a number of features: it is effective, universally applicable, and cheap in terms of training and computational cost. Its effectiveness has been expressed in the thesis statements and concerned the amount of classification mistakes the FSA step is fixing in terms of all considered measures. Especially the amount of invalid scene boundaries (expressed by the IBR measure) has been decreased significantly in all cases.

When considering the proposed methods applicability, it has to be noted that it introduces no additional assumptions besides the continuity of the analyzed video stream. Therefore, any OFA algorithms classifying such data are susceptible to being improved by the FSA approach.

The third listed feature of FSA – its low additional cost when compared to OFA – demonstrates itself in a number of aspects. First, the cost of training is mostly contained in the underlying OFA classifier. This allows to train on image databases and use preexisting OFA implementations, instead of having to create large dedicated video datasets. Only optimizing an FSA algorithm for a specific domain or video source requires a smaller amount of data, which is used to adjust the parameters. What is more, if basic default parameter values are chosen, the FSA step can already be expected to introduce a noticeable improvement of the classification quality. Secondly, the

FSA method (especially in the iFSA variant) introduces only a minimal computational overhead. Therefore there is no additional cost when compared to using a single OFA classifier.

Besides formally defining the FSA method itself, this work also provides an in-depth discussion of the conditions required to make the FSA approach applicable. Namely, the continuity of functions has been generalized into the realm of functions discrete both in time and value domains. Solid criteria have been provided for establishing the continuity of a recording representing a given category. They are compliant with the perceived continuity of real-life videos, which are the actual subject of interest for most OFA applications.

Specific scenarios can be conceived where the applicability of the FSA approach could be questioned. They concern especially domains with a very high cost of false negative classifications and either OFA algorithms with a very small recall rate or video of very low quality. In both such cases the FSA step could possibly cancel single detections. The assumptions under which the FSA method has been defined (sufficient quality of OFA algorithm, continuous video) ensure preventing the application of the FSA method in such scenarios.

Testing environment

In the effort to prove the thesis statements and evaluate the properties of the FSA algorithms, a complex testing environment has been designed and implemented. It executes all stages of the testing procedure - from converting all video streams into a common format, through acquiring the OFA results, training and testing both FSA variants, to evaluating the final results. It concludes all those steps with stored check-points, allowing to restart interrupted computations with a fine granularity.

The created environment allows for a simple inclusion of new methods for testing as well as simplified development. This all comes due to its caching functionality which allows to prevent unnecessarily re-running expensive computations. After running the most time-consuming OFA step, the applied optimizations and storage of intermediate results for all processing stages allow to reduce the time for evaluating new FSA methods from days to minutes (or hours in the case of more complex parameterizations).

Furthermore, multiple available parallelization options have been exploited to use any number of cores available in the runtime system. Parallelization on all levels ensures that the testing procedure can use all of the available computing power. These improvements have been especially valuable for extensive testing, as they made it possible to fully utilize the test environment on a Google Cloud Platform Compute Engine virtual system, whose number of processors is adjustable – allowing to speed-up the computations at will. This also relates to the construction of the FSA algorithms, which easily supports the parallel classification of video segments.

Implementation work

Altogether, for the evaluation of the proposed method, a significant implementation effort was required, which concerned all levels of the testing procedure. All datasets have been converted

to a common format, which allowed to apply distortions on them and pass consecutive frames to the OFA classifier. One of the chosen OFA algorithms, which was provided only in a descriptive form by its authors, had to be recreated for the evaluation in this dissertation. The classifier achieved results corresponding to those claimed by its authors, which allowed it to be included in the testing procedure.

Regarding the FSA method itself, the implementation for the testing procedure included two variants of the FSA method (iFSA, fFSA), three decision functions (including the *model-based scene boundary restriction* decision rule, which is also an original contribution of this work) and all the other building blocks of the FSA method. The structure of the algorithms allowed for a trivial parallelization on multiple levels (low-level: openCV parallelization; frames: OFA classification; time windows: FSA steps; parameter sets: testing procedure implementation), which results in an almost perfect scalability on multi-core processor systems.

Experiments

The experiments performed as a part of this work have verified the usefulness of the FSA method and its low overhead cost. It has been shown that it is applicable in multiple scenarios - both to improve the underlying OFA method, as well as to provide robustness for distortions of the video stream (e.g. rainfall/fog, changes in lighting, flashes of light, blurs, etc.).

All experiments confirmed the thesis statements with only minor exceptions, all of which have been analyzed and discussed. The FSA method improves the OFA classification results, by decreasing the considered error measure values on average by 20%. It performs well when distortions are applied to the video stream, ensuring that the quality of the OFA classification on an undistorted stream is kept. This holds for both analyzed variants of the FSA method.

The relation of the FSA method's effectiveness with the window size, distribution parameter and change acceptance threshold has been established. Also relations between all pairs of parameters have been analyzed and discussed. The quality improvement of the proposed FSA algorithms turned out to be on a predictable and high level. Especially worth noting is the result which compared the performance of the FSA algorithm optimized in terms of three measures with those optimized in terms of single ones of them. It has shown that applying the FSA approach uniformly improves all quality criteria, which confirms its wide applicability.

Future improvements

After having confirmed the thesis statements and analyzed the experiment results in detail, we can specify four categories of directions for further research.

The first kind of issues which can be addressed, is the definition of quality measures expressing the result stability. The MBR measure has been defined as a symmetrical analogue to the IBR (as in the FPR/FNR pair). It did not prove useful during experimentation, as it causes numerical issues and strongly restricts the optimized FSA method. Multiple other approaches of evaluating scene segmentation algorithms can be analyzed again in an effort to define better measures.

Secondly, further work should focus on the selection of image similarity measures, which would both emphasize semantical similarity and perform in an efficient manner. The results of our evaluation indicate that the most computationally efficient methods have to begin with a radical reduction of dimensionality. This can be acquired by preprocessing the compared images and extracting their features, but also by introducing reference frames and utilizing the triangle inequality in similarity distances to approximate the similarity function's value. Another direction of work regarding similarity measures, which could lead to developing new decision rules, is the frame clustering idea described at the end of section 6.5.

A third area for improving and extending the FSA approach are applications with domains other than binary classification. Within this work we have already noted the possibility of including the classification confidence for binary classification into the decision rule reasoning. We have also briefly discussed and proposed a framework for improving multi-categorical classifications. A general scheme can be considered, where the decision function of the FSA step is implemented by a more complex classifier (e.g. an artificial neural network), whose inputs are results acquired for all frames in the window and output is the suggested value for the central frame. Such an approach would need to consider the risk of increasing the, otherwise very low, cost of optimizing the FSA step parameters.

The fourth and last direction for further research is to continue the application and evaluation of the FSA methods in more domains and for more OFA algorithms. Algorithms classifying single frames in videos are ubiquitous in multiple other fields, besides those considered in this work – medical applications being an outstanding example. Furthermore, the own parameters of the OFA algorithms can be considered in such experiments, to choose those most susceptible to the FSA improvement. If possible, such experiments should consider including other OFA-improving methods (global or dVSA) for a thorough comparison.

Overall, the FSA approach has been proven to be an inexpensive and highly efficient improvement option for existing OFA methods. It is widely applicable and does not necessarily introduce additional computational or data-acquisition requirements. Possibilities of further improving it exist and have been identified, for which this dissertation is a significant first step. The deep discussion and exemplary experiments with FSA methods are a reference point for applying FSA improvements to acquired classification results. They constitute also a well proven opening for further investigations in the subject of improving image classifications in videos.

8 Bibliography

[1] S. Abdallah, M. Sandler, C. Rhodes, and M. Casey. Using duration models to reduce fragmentation in audio segmentation. *Machine Learning*, 65(2-3):485–515, nov 2006.

[2] S. Atasoy, D. Mateus, J. Lallemand, A. Meining, G.-Z. Yang, and N. Navab. Endoscopic video manifolds. *Medical Image Computing and Computer-Assisted Intervention*, 13(Pt 2):437–445, 2010.

[3] Australian Bureau of Statistics. *An Introductory Course on Time Series Analysis – Electronic Delivery*. 2005.

[4] M. Badurowicz, T. Cieplak, and J. Montusiewicz. The Cloud Computing Stream Analysis System for Road Artefacts Detection. In *Computer Networks*, pages 360–369. 2016.

[5] M. Badurowicz, T. Cieplak, and J. T. Montusiewicz. On-the-fly community-driven mobile accelerometer data analysis system for road quality assessment. *Applied Computer Science*, 12(4):18–27, 2016.

[6] S. Bai, J. Z. Kolter, and V. Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271*, mar 2018.

[7] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pages 659–663, 1977.

[8] S. Basu and M. Meckesheimer. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems*, 11(2):137–154, aug 2006.

[9] A. Bellet, A. Habrard, and M. Sebban. A Survey on Metric Learning for Feature Vectors and Structured Data. Technical report, jun 2013.

[10] R. Bellman and R. Ernest. *Dynamic programming*. Dover Publications, 2003.

[11] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive Online Analysis. *Journal of Machine Learning Research*, 11(May):1601–1604, 2010.

[12] A. Blokus and H. Krawczyk. Systematic Approach to Binary Classification of Images in Video Streams using Shifting Time-Windows. *Signal, Image and Video Processing*, ((Accepted for publication)), 2018.

[13] A. Blokus and H. Krawczyk. Improving Traffic Light Recognition Methods using Shifting Time-Windows. In *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–5, Maribor, jun 2018. IEEE.

[14] A. Blokus and H. Krawczyk. Improving methods for detecting people in video recordings using shifting time-windows. In X. Jiang and J.-N. Hwang, editors, *Tenth International Conference on Digital Image Processing (ICDIP 2018)*, volume 10806, page 121, Shanghai, aug 2018. SPIE.

[15] A. Blokus and H. Krawczyk. Impact of shifting time-window post-processing on the quality of face detection algorithms. In *2018 11th International Conference on Human System Interaction (HSI)*, pages 77–83. IEEE, jul 2018.

[16] A. Blokus, A. Brzeski, J. Cychnerski, T. Dziubich, and M. Jędrzejewski. Real-Time Gastrointestinal Tract Video Analysis on a Cluster Supercomputer. In W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, editors, *Complex Systems and Dependability*, volume 170 AISC, pages 55–68. Springer, 2012.

[17] A. Blokus, A. Brzeski, J. Cychnerski, and M. Jędrzejewski. Endoscopic Video Classification with the Consideration of Temporal Patterns. In *Proceedings of the 5th International Interdisciplinary Technical Conference of Young Scientists InterTech 2012*, pages 237–241, Poznań, 2012. Wydawnictwo Politechniki Gdańskiej.

[18] A. Blokus, A. Brzeski, and J. Cychnerski. Issues of classification function continuity in endoscopic video classification. In *Zeszyty Naukowe Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej, Publikacja konferencyjna ICT Young*, 2013.

[19] A. Blokus, J. Cychnerski, and A. Brzeski. Accelerating video frames classification with metric based scene segmentation. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(8):5311–5315, 2014.

[20] Bo Han and Weiguo Wu. Video scene segmentation using a novel boundary evaluation criterion and dynamic programming. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE, jul 2011.

[21] E. Z. Borzeshi, O. P. Concha, M. Piccardi, O. Perez Concha, and R. Y. D. Xu. Joint Action Segmentation and Classification by an Extended Hidden Markov Model. *IEEE Signal Processing Letters*, 20(12):1207–1210, dec 2013.

[22] S. Bourennane and C. Fossati. Comparison of shape descriptors for hand posture recognition in video. *Signal, Image and Video Processing*, 6(1):147–157, mar 2012.

[23] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[24] J. Brankov, M. Wernick, M. King, Y. Yang, and M. Narayanan. Spatially Adaptive Temporal Smoothing for Reconstruction of Dynamic Image Sequences. *IEEE Transactions on Nuclear Science*, 53(5):2769–2777, oct 2006.

[25] J. J. G. Brankov, M. M. N. Wernick, M. M. V. Narayanan, and Y. Yang. Spatially-adaptive temporal smoothing for reconstruction of dynamic and gated image sequences. In *2000 IEEE Nuclear Science Symposium. Conference Record (Cat. No.00CH37149)*, volume 2, pages 15/146–15/150. IEEE, 2000.

[26] A. Brzeski, A. Blokus, and J. Cychnerski. An Overview of Image Analysis Techniques in Endoscopic Bleeding Detection. *International Journal of Innovative Research in Computer and Communication Engineering*, 1 (5), 2013.

[27] M. Burgin. Continuity in Discrete Sets. *arXiv:1002.0036*, jan 2010.

[28] W. Cai, Y. Song, and D. D. Feng. Regression and classification based distance metric learning for medical image retrieval. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1775–1778. IEEE, may 2012.

[29] Y. Cao, D. Liu, W. Tavanapong, J. Wong, J. Oh, and P. C. De Groen. Computer-aided detection of diagnostic and therapeutic operations in colonoscopy videos. *IEEE Transactions on Biomedical Engineering*, 54(7):1268–1279, 2007.

[30] R. D. Charette and F. Nashashibi. Real Time Visual Traffic Lights Recognition with Image Processing. *Advanced Robotics*, (33):358–363, 2009.

[31] Y. Chen and J. Lee. A review of machine-vision-based analysis of wireless capsule endoscopy video. *Diagnostic and therapeutic endoscopy*, 2012:418037, jan 2012.

[32] D. Cox and E. J. Snell. *Analysis of Binary Data, Second Edition.* CRC Press, 1989.

[33] J. Cychnerski, A. Brzeski, A. Blokus, T. Dziubich, and M. Jędrzejewski. Konstrukcja bazy danych dla systemu wspomagania diagnostyki chorób przewodu pokarmowego. In *Studia Informatica*, volume 33, 2012.

[34] J. Cychnerski, A. Brzeski, and A. Blokus. Method of training the endoscopic video analysis algorithms to maximize both accuracy and stability. In *ICT Young 2013*, number 10, 2013.

[35] K. Czuszynski, J. Ruminski, and A. Kwasniewska. Gesture Recognition With the Linear Optical Sensor and Recurrent Neural Networks. *IEEE Sensors Journal*, 18(13):5429–5438, jul 2018.

[36] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv preprint, arXiv:1605.06409*, may 2016.

[37] R. de Charette and F. Nashashibi. Traffic light recognition using image processing compared to learning processes. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 333–338, oct 2009.

[38] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1491–1498. IEEE, sep 2009.

[39] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 613–622, New York, New York, USA, 2001. ACM Press.

[40] M. Ebdelli, O. Le Meur, and C. Guillemot. Video Inpainting With Short-Term Windows: Application to Object Removal and Error Concealment. *IEEE Transactions on Image Processing*, 24(10):3034–3047, oct 2015.

[41] S. Erturk. Image sequence stabilisation: motion vector integration (MVI) versus frame position smoothing (FPS). In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat. No.01EX480)*, pages 266–271. Univ. Zagreb, 2001.

[42] A. Fabijanska and J. Goclawski. The Segmentation of 3D Images Using the Random Walking Technique on a Randomly Created Image Adjacency Graph. *IEEE Transactions on Image Processing*, 24(2):524–537, feb 2015.

[43] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, aug 2013.

[44] J. Felipe, A. Machado Traina, and C. Traina. Global Warp Metric Distance: Boosting Content-based Image Retrieval through Histograms. In *Seventh IEEE International Symposium on Multimedia (ISM'05)*, pages 295–302. IEEE, 2005.

[45] G. Fettweis and H. Meyr. Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck. *IEEE Transactions on Communications*, 37(8):785–790, 1989.

[46] G. Fettweis and H. Meyr. Feedforward architectures for parallel viterbi decoding. *Journal of VLSI signal processing systems for signal, image and video technology*, 3(1-2):105–119, jun 1991.

[47] D. Figueira, M. Taiana, J. C. Nascimento, and A. Bernardino. A Window-Based Classifier for Automatic Video-Based Reidentification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(12): 1736–1747, dec 2016.

[48] G. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[49] B. Froba and C. Kublbeck. Face Tracking by Means of Continuous Detection. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, page 65. IEEE, 2004.

[50] G. Gallo and A. Torrisi. Boosted Wireless Capsule Endoscopy Frames Classification. In *PATTERNS 2011, The Third International Conferences on Pervasive Patterns and Applications*, pages 25–30, 2011.

[51] G. Gallo, E. Granata, and G. Scarpulla. Sudden Changes Detection in WCE Video. In P. Foggia, C. Sansone, and M. Vento, editors, *Image Analysis and Processing – ICIAP 2009*, volume 5716 of *Lecture Notes in Computer Science*, pages 701–710. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[52] G. Gallo, E. Granata, and A. Torrisi. Information Theory Based WCE Video Summarization. In *2010 20th International Conference on Pattern Recognition*, pages 4198–4201. IEEE, 2010.

[53] J. Gama. A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence*, 1(1):45–55, apr 2012.

[54] Y. Gaol, W. Tavanapongl, K. Kim, J. Wong, J. Oh, and P. C. D. Groen. A framework for parsing colonoscopy videos for semantic units. *Gastroenterology And Hepatology*, pages 1879–1882, 2004.

[55] U. Gargi, R. Kasturi, and S. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):1–13, 2000.

[56] J. M. Gauch, S. Gauch, S. Bouix, and X. Zhu. Real time video scene detection and classification. *Information Processing & Management*, 35(3):381–400, 1999.

[57] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2141–2148. IEEE, jun 2010.

[58] Guodong Guo, Hong-Jiang Zhang, and S. Li. Distance-from-boundary as a metric for texture image retrieval. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 3, pages 1629–1632. IEEE, 2001.

[59] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier Detection for Temporal Data tutorial. In *2013 SIAM International Conference on Data Mining*, Austin, Texas, USA, 2013.

[60] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, sep 2014.

[61] A. Hadid and M. Pietikainen. From still image to video-based face recognition: an experimental analysis. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 813–818. IEEE, 2004.

[62] M. Häfner, M. Liedlgruber, A. Uhl, A. Vécsei, and F. Wrba. Color treatment in endoscopic image classification using multi-scale local color vector patterns. *Medical Image Analysis*, 16(1):75–86, 2012.

[63] O. Haji-Maghsoudi, A. Talebpour, H. Soltanian-Zadeh, and N. Haji-maghsoodi. Automatic organs' detection in WCE. In *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, pages 116–121. IEEE, may 2012.

[64] A. E. Hassanien, A. Abraham, J. F. Peters, G. Schaefer, and C. Henry. Rough sets and near sets in medical imaging: a review. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 13(6):955–68, nov 2009.

[65] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, jun 2016.

[66] D. J. Hill and B. S. Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software*, 25(9):1014–1022, sep 2010.

[67] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, nov 1997.

[68] X.-S. Hua, D. Zhang, M. Li, and H.-J. Zhang. Performance Evaluation Protocol for Video Scene Detection Algorithms. In *Workshop on Multimedia Information Retrieval, in conjunction with 10th ACM Multimedia*, 2002.

[69] J. Huang, Z. Liu, and Y. Wang. Joint video scene segmentation and classification based on hidden Markov model. In *2000 IEEE International Conference on Multimedia and Expo*, volume 3, pages 1551–1554. IEEE, 2000.

[70] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[71] G. Iddan, G. Meron, A. Glukhovsky, and P. Swain. Wireless capsule endoscopy. *Nature*, 405(6785):417, may 2000.

[72] Indrabayu, R. Y. Bakti, I. S. Areni, and A. A. Prayogi. Vehicle detection and tracking using Gaussian Mixture Model and Kalman Filter. In *2016 International Conference on Computational Intelligence and Cybernetics*, pages 115–119. IEEE, 2016.

[73] G. Iyengar and A. Lippman. Models for automatic classification of video sequences. In *SPIE Proc. Storage and Retrieval for Image and Video Databases*, pages 216–227, 1997.

[74] D. Jacobs, D. Weinshall, and Y. Gdalyahu. Condensing image databases when retrieval is based on non-metric distances. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 596–601. Narosa Publishing House, 1998.

[75] H. V. Jagadish, N. Koudas, and S. Muthukrishnan. Mining Deviants in a Time Series Database. In *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 102–113. Morgan Kaufmann Publishers Inc., sep 1999.

[76] Joe Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702. IEEE, jun 2015.

[77] J. F. Kaiser and W. A. Reed. Data smoothing using low-pass digital filters. *Review of Scientific Instruments*, 48(11):1447, 1977.

[78] S. Kamkar and R. Safabakhsh. Vehicle detection, counting and classification in various conditions. *IET Intelligent Transport Systems*, 10(6):406–413, aug 2016.

[79] A. Karargyris and N. Bourbakis. Wireless Capsule Endoscopy and Endoscopic Imaging: A Survey on Various Methodologies Presented. *IEEE Engineering in Medicine and Biology Magazine*, 29(1):72–83, 2010.

[80] B. Kedem and K. Fokianos. Regression Theory for Categorical Time Series. *Statistical Science*, 18(3): 357–376, aug 2003.

[81] B. Kedem and K. Fokianos. *Regression Models for Time Series Analysis*. 2005.

[82] B. Kedem and K. Fokianos. Regression Models for Binary Time Series. In *Modeling Uncertainty*, pages 185–199. Kluwer Academic Publishers, Boston, 2005.

[83] Y. Keller and A. Averbuch. Multisensor image registration via implicit similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):794–801, may 2006.

[84] V. S. Kodogiannis and M. G. Boulougoura. An Adaptive Neurofuzzy Approach for the Diagnosis in Wireless Capsule Endoscopy Imaging. *International Journal of Information Technology*, 13(1):46–56, 2007.

[85] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough Sets: A Tutorial, 1998.

[86] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, jan 2001.

[87] H. Krawczyk and J. Proficz. KASKADA – MULTIMEDIA PROCESSING PLATFORM ARCHITECTURE. In *SIGMAP Conference Proceedings*, 2010.

[88] H. Krawczyk and J. Proficz. Real-Time Multimedia Stream Data Processing in a Supercomputer Environment. In *Interactive Multimedia*. InTech, mar 2012.

[89] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[90] J. Lee, J. Oh, S. K. Shah, X. Yuan, and S. J. Tang. Automatic classification of digestive organs in wireless capsule endoscopy videos. *Proceedings of the 2007 ACM symposium on Applied computing SAC 07*, (c): 1041–1045, 2007.

[91] B. Y. Li, A. S. Mian, W. Liu, and A. Krishna. Using Kinect for face recognition under varying poses, expressions, illumination and disguise. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 186–192. IEEE, jan 2013.

[92] Y. Li and R. L. Stevenson. A Similarity Metric for Multimodal Images Based on Modified Hausdorff Distance. In *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pages 143–148. IEEE, sep 2012.

[93] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen. Face Video Retrieval With Image Query via Hashing Across Euclidean Space and Riemannian Manifold. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4758–4767, 2015.

[94] M. Liedlgruber and A. Uhl. Computer-aided decision support systems for endoscopy in the gastrointestinal tract: a review. *IEEE reviews in biomedical engineering*, 4:73–88, 2011.

[95] S. Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, sep 2010.

[96] T. Lin and H.-j. Zhang. Automatic video scene extraction by shot grouping. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 4:39–42, 2000.

[97] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. *arXiv preprint, arXiv:1708.02002*, aug 2017.

[98] H. Ling and K. Okada. EMD-L 1: An Efficient and Robust Algorithm for Comparing Histogram-Based Descriptors. *Lecture Notes in Computer Science*, 3953:330–343, 2006.

[99] X. Ling, L. Chao, L. Huan, and X. Zhang. A General Method for Shot Boundary Detection. In *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, pages 394–397. IEEE, 2008.

[100] P. Liskowski and K. Krawiec. Segmenting Retinal Blood Vessels With Deep Neural Networks. *IEEE Transactions on Medical Imaging*, 35(11):2369–2380, nov 2016.

[101] D. Liu and T. Chen. Object Detection in Video with Graphical Models. In *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, volume 5, pages 693–696. IEEE, 2006.

[102] M. Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1696–1703, 2010.

[103] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *ECCV*, pages 21–37. Springer, Cham, oct 2016.

[104] Y. Liu, L. Zeng, and Y. Huang. An efficient HOG–ALBP feature for pedestrian detection. *Signal, Image and Video Processing*, 8(S1):125–134, dec 2014.

[105] Y. Liu, S. Zhang, M. Xu, and X. He. Predicting Salient Face in Multiple-Face Videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3224–3232. IEEE, jul 2017.

[106] B. Lovell and P. Kootsookos. Evaluation of HMM training algorithms for letter hand gesture recognition. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795)*, pages 648–651. IEEE, 2003.

[107] Y. Lu, C. Lu, and C.-K. Tang. Online Video Object Detection Using Association LSTM. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2363–2371. IEEE, oct 2017.

[108] J. Ma and S. Perkins. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1741–1745. IEEE, 2003.

[109] M. Mackiewicz. Capsule Endoscopy - State of the Technology and Computer Vision Tools After the First Decade. In O. Pascu and A. Seicean, editors, *New Techniques in Gastrointestinal Endoscopy*, chapter 7, pages 103–124. InTech, oct 2011.

[110] M. Mackiewicz, J. Berens, and M. Fisher. Wireless capsule endoscopy color video segmentation. *IEEE Transactions on Medical Imaging*, 27(12):1769–1781, 2008.

[111] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, apr 1997.

[112] G. D. Magoulas. Neuronal networks and textural descriptors for automated tissue classification in endoscopy. *Oncology reports*, 15 Spec no:997–1000, 2006.

[113] G. D. Magoulas, V. P. Plagianakos, and M. N. Vrahatis. Neural network–based colonoscopic diagnosis using on–line learning and differential evolution. *Applied Soft Computing*, 4(4):369–379, sep 2004.

[114] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised Video Summarization with Adversarial LSTM Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2982–2991. IEEE, jul 2017.

[115] P. Marchand and L. Marmet. Binomial smoothing filter: A way to avoid some pitfalls of least-squares polynomial smoothing. *Review of Scientific Instruments*, 54(8):1034, 1983.

[116] M. Q.-H. Meng and B. Li. Tumor CE image classification using SVM-based feature selection. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1322–1327. IEEE, oct 2010.

[117] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[118] G. D. F. Morales and A. Bifet. SAMOA: Scalable Advanced Massive Online Analysis. *Journal of Machine Learning Research*, 16(Jan):149–153, 2015.

[119] B. Munzer, K. Schoeffmann, and L. Boszormenyi. Detection of Circular Content Area in Endoscopic Videos for Efficient Encoding and Improved Content Analysis. Technical report, Institute of Information Technology, University Klagenfurt, 2012.

[120] B. Munzer, K. Schoeffmann, and L. Boszormenyi. Detection of circular content area in endoscopic videos. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 534–536. IEEE, jun 2013.

[121] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7):971–987, jul 2002.

[122] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7):971–987, 2002.

[123] G. Pan and L. Wang. Swallowable Wireless Capsule Endoscopy: Progress and Technical Challenges. *Gastroenterology Research and Practice*, 2012.

[124] Z. Pawlak. On Some Issues Connected With Roughly Continuous Functions. 1995.

[125] Z. Pawlak. Rough calculus. Technical Report 58, 1995.

[126] W. Pei, T. Baltrušaitis, D. M. J. Tax, and L.-P. Morency. Temporal Attention-Gated Model for Robust Sequence Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 820–829. IEEE, dec 2017.

[127] Y. Poleg, T. Halperin, C. Arora, and S. Peleg. EgoSampling: Fast-Forward and Stereo for Egocentric Videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4768–4776, 2015.

[128] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[129] V. P. Rainer Lienhart, Er Kuranov. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In *In DAGM 25th Pattern Recognition Symposium*, pages 297–304, 2003.

[130] M. Ramona, G. Richard, and B. David. Vocal detection in music with support vector machines. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1885–1888. IEEE, mar 2008.

[131] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. IEEE, dec 2017.

[132] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, jun 2016.

[133] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, jun 2017.

[134] F. Riaz, F. B. Silva, M. D. Ribeiro, and M. T. Coimbra. Invariant Gabor texture descriptors for classification of gastroenterology images. *IEEE transactions on bio-medical engineering*, 59(10):2893–904, oct 2012.

[135] G. Q. Rosa Ruiloba, Stephane March. Towards a Standard Protocol for the Evaluation of Video-to-Shots Segmentation Algorithms. In *First European Workshop on Content-Based Multimedia Indexing*, 1999.

[136] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE, nov 2011.

[137] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[138] D. B. Russakoff, C. Tomasi, T. Rohlfing, and C. R. Maurer. Image Similarity Using Mutual Information of Regions. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, pages 596–607, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[139] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (3rd Edition)*. Prentice Hall, 2009.

[140] U. Sakarya and Z. Telatar. Video scene detection using graph-based representations. *Signal Processing: Image Communication*, 25(10):774–783, nov 2010.

[141] V. Stanisavljevic, Z. Kalafatic, and S. Ribaric. Optical flow estimation over extended image sequence. In *2000 10th Mediterranean Electrotechnical Conference. Information Technology and Electrotechnology for the Mediterranean Countries. Proceedings. MeleCon 2000 (Cat. No.00CH37099)*, volume 2, pages 546–549. IEEE, 2000.

[142] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Estimating 3D hand pose using hierarchical multi-label classification. *Image and Vision Computing*, 25(12):1885–1894, dec 2007.

[143] R. Sukthankar and M. Hebert. Efficient visual event detection using volumetric features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 166–173 Vol. 1. IEEE, 2005.

[144] J. Sun, J. Wang, and T.-C. Yeh. Video Understanding: From Video Classification to Captioning. Technical report, Stanford University, 2017.

[145] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *AAAI Conference on Artificial Intelligence*, San Francisco, feb 2017.

[146] M. Teutsch and W. Kruger. Robust and fast detection of moving vehicles in aerial videos using sliding windows. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 26–34. IEEE, jun 2015.

[147] That Mon Htwe, Chee Khun Poh, Liyuan Li, Jiang Liu, Eng Hui Ong, and Khek Yu Ho. Vision-based techniques for efficient Wireless Capsule Endoscopy examination. In *2011 Defense Science Research Conference and Expo (DSR)*, pages 1–4. Department of Computer Vision and Image Understanding, Institute for Infocomm Research, Singapore 138632, IEEE, aug 2011.

[148] K.-L. Ton-Thi, T.-A. Nguyen, and M.-C. Hong. Video stabilization algorithm using a moving alpha-trimmed mean filter window. In *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, pages 1–2. IEEE, jun 2014.

[149] G. V. Trunk. A Problem of Dimensionality: A Simple Example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):306–307, jul 1979.

[150] S. Tsevas, D. K. Iakovidis, D. Maroulis, and E. Pavlakis. Automatic frame reduction of Wireless Capsule Endoscopy video. In *2008 8th IEEE International Conference on BioInformatics and BioEngineering*, pages 1–6. IEEE, 2008.

[151] P. Turaga. *Statistical and Geometric Modeling of Spatio-Temporal Patterns for Video Understanding*. PhD thesis, University of Maryland, 2009.

[152] K. Uosaki and P. Statement. Integer programming approach to optimal smoothing of two-state Markov sequences. In *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 1661–1664. Institute of Electrical and Electronics Engineers, 1986.

[153] J. Vendrig and M. Worring. Systematic evaluation of logical story unit segmentation. *IEEE Transactions on Multimedia*, 4(4):492–499, dec 2002.

[154] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518. IEEE Comput. Soc, 2001.

[155] P. Viola and M. Jones. Robust Real-time Object Detection. *International Journal of Computer Vision*, 2001.

[156] P. Viola and W. M. Wells III. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[157] H. Vu, T. Echigo, R. Sagawa, K. Yagi, M. Shiba, K. Higuchi, T. Arakawa, and Y. Yagi. Contraction detection in small bowel from an image sequence of wireless capsule endoscopy. *Medical Image Computing and Computer-Assisted Intervention*, 10(Pt 1):775–783, 2007.

[158] H. Wang, S. Zhang, W. Liang, F. Wang, and Y. Yao. Content-based image retrieval using fractional distance metric. In *2012 International Conference on Image Analysis and Signal Processing*, pages 1–5. IEEE, nov 2012.

[159] X. Wang and X.-P. Zhang. An ICA Mixture Hidden Conditional Random Field Model for Video Event Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):46–59, jan 2013.

[160] Y. Wang, X. Xu, and Z. Yu. Notes for rough derivatives and rough continuity in rough function model. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, pages 245–247. IEEE, aug 2010.

[161] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, apr 2004.

[162] J. Weber, S. Lefevre, and P. Gancarski. Video Object Mining: Issues and Perspectives. In *2010 IEEE Fourth International Conference on Semantic Computing*, pages 85–90. IEEE, sep 2010.

[163] Y. Wei, S. M. Bhandarkar, and K. Li. Semantics-Based Video Indexing using a Stochastic Modeling Approach. In *2007 IEEE International Conference on Image Processing*, volume 4, pages IV – 313–IV – 316. IEEE, 2007.

[164] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *CVPR 2011 Workshops*, pages 74–81. IEEE, jun 2011.

[165] S. Wu and J. Yang. Local Image Distance Metric Learning. In *2010 Chinese Conference on Pattern Recognition (CCPR)*, pages 1–5. IEEE, oct 2010.

[166] L. Xie, I. B. M. T. J. Watson, and S.-f. Chang. Pattern Mining in Visual Concept Streams. In *2006 IEEE International Conference on Multimedia and Expo*, number 1, pages 297–300. IEEE, jul 2006.

[167] F. Xiong, X. Shi, and D.-Y. Yeung. Spatiotemporal Modeling for Crowd Counting in Videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5161–5169. IEEE, oct 2017.

[168] W. Xiong and J. C.-M. Lee. Efficient Scene Change Detection and Camera Motion Annotation for Video Classification. *Computer Vision and Image Understanding*, 71(2):166–181, aug 1998.

[169] Y. Yang, B. C. Lovell, and F. Dadgostar. Content-Based Video Retrieval (CBVR) System for CCTV Surveillance Videos. In *2009 Digital Image Computing: Techniques and Applications*, pages 183–187. IEEE, 2009.

[170] J. Yu, J. Amores, N. Sebe, and Q. Tian. A New Study on Distance Metrics as Similarity Measurement. In *2006 IEEE International Conference on Multimedia and Expo*, pages 533–536. IEEE, jul 2006.

[171] J. Zhang. An improved clustering for action recognition in online video. In *2011 International Conference on Multimedia Technology*, pages 180–183. Ieee, jul 2011.

[172] Z. Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE Multimedia*, 19(2):4–10, feb 2012.

[173] Q. Zhao and M. Q. Meng. An abnormality based WCE video segmentation strategy. In *2010 IEEE International Conference on Automation and Logistics, ICAL 2010*, pages 565–570. IEEE, aug 2010.

[174] Q. Zhao and M. Q.-H. Meng. WCE video abstracting based on novel color and texture features. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 455–459. IEEE, dec 2011.

Glossary

Notation

The mathematical symbols notation follows the following general patterns:

- capital symbols: sets, series, constants, random variables;
- small symbols: singular/vector values and variables;
- symbols of functions: as above, according to return value;
- others: according to common convention;

- function domains:
  - specified where suitable, omitted for informally defined functions
  - simplified notation: for a given $f : A \to B$ and if $C \subseteq A$ we assume that $f(C) = \{f(c) : c \in C\}$.

Symbol	**Definition**
$\equiv$	equivalence
$\circ$	function composition - $(f \circ g)(x) = f(g(x))$
$\#X$	number of elements in X ($\#X = \sum_{x \in X} 1$)
$c : \Theta \to \mathcal{C}$	function c whose domain is Θ codomain is $\mathcal{C}$
f	exemplary function
f'	first order derivative of f
$\mathbb{N}_+$	Natural positive numbers set
$\mathbb{N}$	Natural numbers (with zero)
$\mathbb{R}_{\geq 0}$	Non-negative real numbers set
$\binom{n}{k}$	binomial coefficient, *"n choose k"*
$[i \dots j]$	sequence of integer values from i to j (inclusive)
$\{\cdot\}$	set
$(a, b, c, \dots)$	sequence
$\ll$	significantly less than
$P(\cdot)$	probability of a given event
$\lvert \cdot \rvert$	absolute value
$O(\cdot)$	Big O asymptotic notation
Θ	set of all possible images

D	size of images
$\mathbb{F}$	set of feature vectors
$\mathcal{C}$	set of all possible classifications in given problem. Depending on problem, for binary classifications $\mathcal{C} = [0;1]$, $\mathcal{C} = \{0,1\}$, for numerical properties $\mathcal{C} = \mathbb{R}$, $\mathcal{C} = \mathbb{Z}$, $\mathcal{C} = \mathbb{N}$
$\phi : \Theta \to \mathbb{F}$	a feature Vector Function – a function summarizing an image into a vector of features (numerical properties)
κ	a classifying function, $\kappa : \mathbb{F} \to \mathcal{C}$
$\mathcal{T}$	discrete time dimension of the video
X, Y	exemplary metric spaces
$d_X(\cdot,\cdot)$	a metric function in the X metric space
M_d	maximal possible value of metric d
g-continuity	a generalized form of continuity, applicable for discrete functions. Defined on page 31
L	Lipschitz constant for a Lipschitz continuous function
$\mathcal{M}$	frame indices sequence in video
$\mathcal{Z}$	scene length random variable
$step$	time step between consecutive frames ($step = \frac{1}{\text{framerate}}$)
$t_m = m \cdot step$	point in time for frame $m \in \mathcal{M}$
$V(\cdot)$	video recording function, returning pictures of real-life views
Q	representation inaccuracy (difference between real-life view and video frame)
r	frame difference coefficient, $r = L \cdot step + Q$
$p_m = V(v_{m \cdot step})$	frame at discrete point in time m
$w = 2k + 1$	shifting window size
$w_{\max}$	maximal considered window size
$R_{c/g}$	Ratio of classification value's c (0/1) for given ground truth value g (0/1).
B	boundary tolerance
n	length of video recording (number of frames)

Arrays, sequences:

F,F_m	frame sequence, frame at index $m \in \mathcal{M}$
G,G_m	ground truth, ground truth for frame index $m \in \mathcal{M}$
O, O_m	sequence of OFA classifications, OFA classification for frame index $m \in \mathcal{M}$
C, C_m	sequence of classifications (especially - FSA output), classification for frame index $m \in \mathcal{M}$
K, K_m	classification confidence/certainty for frame index $m \in \mathcal{M}$

Terminology and abbreviations

Name	**Definition**
static features	properties detectable on single frames (e.g. blood in a medical picture, presence of faces in a CCTV frame)
dynamic features	properties detectable only in sequences of frames (e.g. velocity of objects, heart rate, peristalsis properties)
continuous video	a video stream representing gradually appearing changes. In real life video this corresponds to single-shot videos.
Lipschitz continuity	a type of continuity, more restrictive than the typical continuity. See Equation 3.2
WCE	Wireless Capsule Endoscopy
ground truth	the actual sequence of labels/classifications of frames in the video sequence
scene	a sequence of consecutive frames presenting a consistent view or motion, which share the same ground truth classification
scene boundary	boundary between two consecutive scenes
view	the observed and recorded visible area or space
(shifting) window	the considered neighborhood of a frame, consisting of a number of consecutive frames from the video. Within this work the considered windows are symmetrical (i.e. the frame of interest is in their center), unless specified otherwise.
distortion	an internal or external factor, which influences the quality of the video
distortion intensity	a measure of the level in which a distortion influenced the quality of the video. In this work expressed on a scale of 0-100%
OFA, One Frame Analysis	a class of methods classifying videos only on a per-frame basis, without state.

FSA, Frame Sequence Analysis	an approach and class of methods based on improving OFA classifications with additional reasoning based on the temporal properties of the stream and classification sequence.
dVSA, Direct Video Segment Analysis	a method which classifies a singe frame based on its whole surrounding video segment
iFSA, Indirect Frame Sequence Analysis	an FSA variant which improves preliminary OFA classifications with a shifting window approach
fFSA, Full Frame Sequence Analysis	an FSA variant introducing into its reasoning both preliminary OFA classifications and properties of the video segment containing the classified frame
result rationalization	the process of changing preliminary results so that they correspond more closely to the expected model of the observed process the step of a fFSA and iFSA method after having acquired the preliminary OFA classifications, FSA step
FPR, FNR	False Positive/Negative Ratio
IBR_B, MBR_B	Invalid/Missing Boundary Ratio with scene boundary tolerance B
RMS	root mean square (in the context of this work: of FPR and FNR)
L^p norm	a norm expressed as the pth root of the sum of pth powers of vector elements
rank aggregation	an approach for ranking records based on multiple criteria/measures, which starts by assigning ranks to the results in terms of each measure
binary classifier	a function assigning one of two classes to its input. Possibly, can return also a value representing the confidence of the assignment.
metric (pseudometric, semimetric)	a function defining the distance between two elements. See definition Definition 4 on page 54.
outlier	a sequence element which stands out in respect to a given characteristic of that sequence. Discussed in subsection 2.2.5
partial order	a (reflexive, antisymmetric, and transitive) relation of items, in which some pairs can remain incomparable

SD, SP,HD,HD_k	Simple Distance, Simple Processed image Distance, Histogram Distance, Histogram distance with k bins
HSV	Hue/Saturation/Value color space
ROC	Receiver Operating Characteristic curve, the relation between the true positive and false positive rates of a binary classifier with a varying classification threshold
HMM	Hidden Markov model
FPS	Frames per Second
sublinear	$f(x)$ is sublinear if $f(x+y) < f(x) + f(y)$
superlinear	$f(x)$ is superlinear if $f(x+y) > f(x) + f(y)$
video segment	a contiguous sequence of frames in a video recording, eg. contents of a shifting window