Co-funded by the Horizon 2020
programme of the European Union

h2020mirror.eu

MIRROR

Migration-Related Risks caused by
misconceptions of Opportunities and Requirements

Grant Agreement No. GA832921

Deliverable D5.1

Work-package	WP5: Multimedia Analysis Methods for Social and Traditional Media
Deliverable	D5.1: First Release of Multimedia Analysis Technologies
Deliverable Leader	CERTH
Quality Assessor	LUH
Dissemination level	Public
Delivery date in Annex I	M12; May 31, 2020
Actual delivery date	M12; May 31, 2020
Revisions	v1.0 Document revised after QA comments
Status	Final
Keywords	Multimedia, Visual Analysis, ASR, Deep Learning, Image/Video Annotation, Video Captioning, Migration-Related Semantic Concepts (MRSCs) Annotation, Visual Sentiment Analysis, Polarity Classification, Language Models

Disclaimer

List of Authors

Partner Acronym	Authors
CERTH	Alexandros Pournaras, Nikolaos Gkalelis, Apostolos Konstantinou, Damianos Galanopoulos, Vasileios Mezaris
SAIL	Gerhard Backfried, Erinç Dikici

Table of Contents

List of Figures

List of Tables

Executive summary

This document describes the first version of the MIRROR methods for multimedia content annotation and automatic speech recognition and also describes a preliminary approach for visual sentiment analysis.

The first section of the document introduces the reader to the objectives of WP5 as well as its positioning to the whole MIRROR architecture and scenarios. The methods developed in WP5 comprise the Audio-visual Media Analysis container (Firure 1) and are responsible for the analysis of the image,video and audio content collected by the Data Manager. The enrichment of such data aims to provide additional insights to the MIRROR users. No issues regarding sensitive data protection arise from the presented methods. The section also includes the structure of the deliverable and the history of the document.

In Section 2, the image/video annotation methods are described in detail. These consist of three separate methods, addressing three different annotation problems: generic visual concept detection (Section 2.1), video captioning (Section 2.2) and the detection of migration-related semantic concepts (Section 2.3). Each presented method produces a different kind of visual annotations and contributes in its own way to multimedia retrieval within the MIRROR system. Concerning visual concept detection, good results were recently achieved thanks to the use of powerful neural network models and large-scale datasets. For instance, models utilizing learnable pooling techniques have provided the best results in the YouTube-8m (YT8M) challenge dataset, which is the largest multilabel video dataset publicly available. To this end, we extend our previous video classification method by introducing a NetVlad and a Mixture of Experts layer in the input and output of our basic architecture, respectively. In this way, nonlinearities in the feature space are removed effectively and we are able to learn a more discriminant video representation. The efficacy of the proposed approach is shown in the experimental evaluation, where an absolute performance gain of 2.9% is obtained over our previous method in the very challenging YT8M dataset. Current state of the art in video captioning approaches (Section 2.2) the problem through detection of concepts using encoder-decoder architecture and by using multiple modalities such as audio, video and speech recognition. The proposed method uses a hierarchical approach of multiple modalities, extracting local and global features. Experiments were conducted on a benchmark dataset using various setups such as introducing optical flow, adding an average pooling of all features, or ignoring the audio modality when audio was missing. The Migration-Related Semantic Concept (MRSC) detection problem (Section 2.3) was addressed as a special type of cross-modal retrieval. For this, we adopted an Ad-hoc Video Search (AVS) method, customized for MIRROR's needs. Each MRSC is manually augmented with sentences for further describing it, while textual and visual content are used to train an attention-based dual encoding network. Finally, since no proper evaluation dataset for the MSRCs detection problem exists, we used datasets from TRECVID's Semantic Indexing (SIN) task, for evaluation and comparison purposes. Our method achieved adequate results compared to methods especially designed for the SIN task, which (contrary to our method) are trained on abundant ground-truth annotated data. Lastly, a REST service has been implemented to serve as the API of the aforementioned methods, handling both images and videos. The service is running is CERTH's infrastructure and is remotely accessed by the core MIRROR system.

Section 3 describes the first steps that have been made towards image sentiment analysis. Extracting sentiment from images and videos complements the text-based sentiment analysis, which is a key task in the detection of media misperceptions. The state-of-the-art in this problem has shifted from using hand-crafted features based on psychology theories to deep learning methods. A deep learning-based feature extractor, coupled with a neural network classifier, has been tested for adjective-noun-pair (ANP) and polarity classification, showing promising initial results. The section concludes by discussing the next steps for improving the performance of image/video sentiment analysis.

Section 4 provides an overview of the automatic speech recognition technology. The amount of multimedia and audio-content has been increasing massively over the past decades and some of the topics encountered in the audio are related to aspects of migration. Communication takes place in a variety of languages and dialects, using different language-registers and often specific terminology and phraseology. In order to produce accurate transcripts which can serve as the basis for enrichment and further downstream technologies, the ASR component employed to transcribe this content must be based on models accurately reflecting them. Work on ASR in MIRROR during the first year of the project focused on the adaptation and extension of existing models to the domains relevant to MIRROR. Furthermore, as adaptation and extension are foreseen to be continuous and repeated tasks (rather than taking place once only), a semi-automatic method for model building has been devised and an environment has been implemented. This environment allows to adapt models efficiently and is foreseen to evolve further within the next project year.

Finally, in Section 5 brief concluding remarks are given, summarizing the progress thus far and the plans for future work in WP5.

1 Introduction

1.1 Objectives and positioning in the MIRROR system architecture and scenarios

WP5 develops a set of multimedia analysis technologies for the annotation and summarization of social and traditional media. Multimedia content in the form of video, audio and images is prevalent in both traditional and social media and plays a vital role in people's communication and information exchange on many topics, including migration. Extracting information from these sources by transcribing and annotating them can help their effective retrieval as the amount of such data increases. Also, automatic summarization of large multimedia collections can assist the end-user's understanding of the data.

Figure 1 presents a simplified view of the full MIRROR system architecture and the positioning of the Audio-Visual Media Analysis container whithin the system. The Audio-Visual Media Analysis container represents the technologies developed in WP5. The Message Handler is the system's core component responsible for orchestrating the framework. The Message Handler feeds the Audio-Visual Media Analysis container with the appropriate data, which are collected in the system's Media Storage component. More precisely, multimedia items, such as videos and images, are fed to the Audio-Visual Media Analysis container that is then responsible for the transcription, annotation and (later) summarization of those items. The metadata produced by the Audio-Visual Media Analysis container complement and enrich the original multimedia data in the storage. More details about the overall MIRROR system design and implementation and the framework's workflow can be found in Deliverable D7.1 [Gallo et al., 2019]. The work in WP5 is parallel to the work in WP4, which focuses on the analysis of text data.

In the MIRROR scenarios, border agencies and migration policy makers, with the help of MIRROR, are monitoring social media for misinformation campaigns targeting potential migrants. By utilizing ASR, transcriptions can be produced from video and audio data. Moreover, analyzing and annotating multimedia, in addition to text, will offer additional insights on the methods used to manipulate migrant opinion. Furthermore, the MIRROR system gives them an idea of how pictures and videos in various media influence the perception of Europe and how biased this portayal is, compared to the reality.

1.2 General data protection considerations

The methods and technologies developed in WP5 and described in this document process only data collected by the Data Manager container (Figure 1). The Data Manager container follows the ”privacy by design” principle, separating the storage of personal information and the storage of data to be analysed, with the purpose of limiting the access to personal data only to the authorised subjects. The output of the analysis performed by the Visual Media Annotation and Sentiment Analysis component does not in any way extract or identify personal information from the image/video data. Any names or other personal data transcribed from audio/video sources by the Automatic Speech Recognition component are substituted by fictious ones by the Pseudonymiser component, so no real names are exposed. More details about the data protection considerations of the Data Manager and the Pseudonymiser component can be found in Deliverable D7.1 [Gallo et al., 2019].

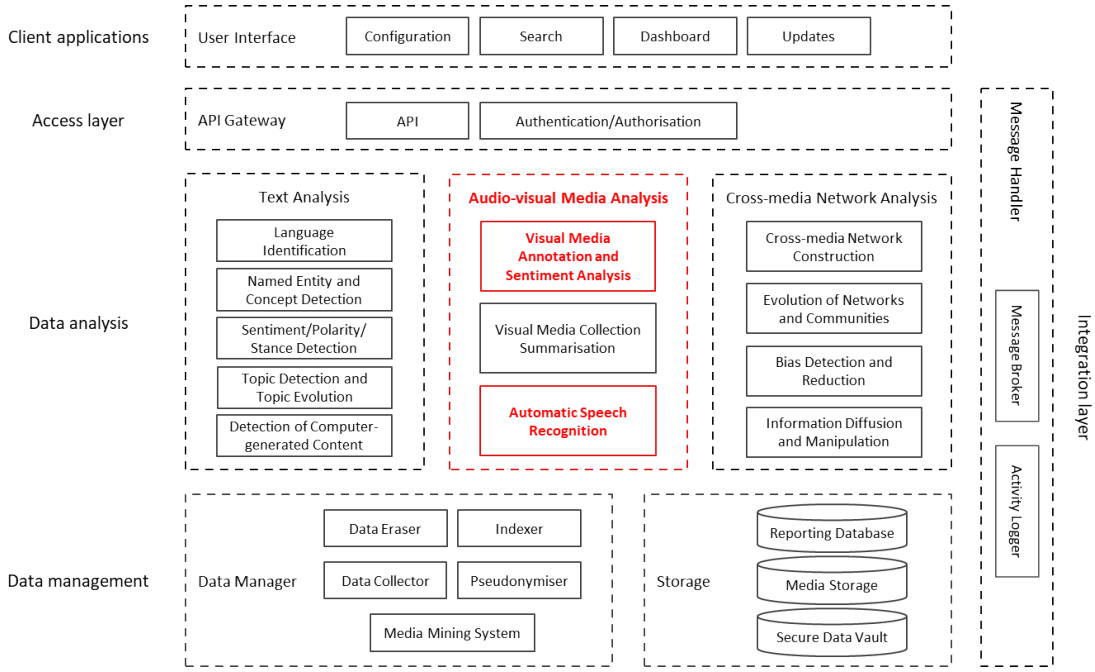

Figure 1: MIRROR system architecture overview. The red rectangles indicate the container and its constituent components that are described in the present document.

1.3 Structure of the Deliverable

Section 2 outlines the methods developed in Task 5.1 that is responsible for the multimedia content annotation technologies. Section 3 deals with the visual sentiment analysis technologies, that are developed in Task 5.2 and Section 4 presents the work carried out in Task 5.4, which deals with the problem of the automatic speech recognition.

1.4 History of the document

Date	Version
24/3/2020	v0.1: Toc Ready
20/4/2020	v0.2: Introduction composed
4/5/2020	v0.3: Section 4 integrated
11/5/2020	v0.4: Section 2,3 and 5 integrated
12/5/2020	v0.5: First complete draft ready
15/5/2020	v0.6: Document ready for QA
28/5/2020	v1.0: Document revised after QA comments

2 Multimedia content annotation

The visual analysis methods described in this section are part of the Visual Media Annotation and Sentiment component (MAS), one of the three components of the Audio-Visual Media Analysis container. There are three types of annotations that are produced in MIRROR for image/video items: generic visual concepts, captions and migration-related semantic concepts. The details on the design of each one of these annotation methods are presented in the following subsections 2.1, 2.2 and 2.3.

All annotations for videos will be provided on a temporal fragment level, whether that is a scene, shot or a sub-shot. Scenes are semantically and temporally coherent segments that correspond to the story-telling parts of the video. Shots are shorter fragment that correspond to sequences of frames captured uninterruptedly by a single camera. Finally, sub-shots are sub-parts of a shot with visually discrete content. For temporal fragmentation, the methods of [Gygli, 2017] and [Apostolidis et al., 2018] are adopted.

Generic visual concepts pools consist of labels corresponding to objects, places, animals, activities etc that can be visually recognized in an image or video. These visual concepts can not predict the thematic context of, e.g., an interview, but can only recognize what can be visually seen, e.g. a human, the background etc. This type of concepts can be very useful when a user wants to retrieve images or video segments that show specific visual items such as church or boat, especially when the image/video collection is huge.

Captions are short sentences that describe an image or a video. Captions are more complicated than visual concepts since they describe an action happening, or a state. Similarly with the visual concepts, caption text can be used to retrieve relevant multimedia content by means of a simple search query.

Migration-related semantic concepts (MRSCs) are high-level concepts relevant to issues concerning migration. Extracting MRSCs from images/videos is a very challenging task since many of the high-level concepts associated with migration, such as ”economics” and ”political factors”, are difficult to be derived from the low-level visual features of images and videos. Nevertheless, to the extent possible, our method will predict the most relevant MRSCs of a given image or video fragment by analyzing its visual content.

2.1 Deep learning for efficient visual classification

2.1.1 Problem statement

On YouTube, 500 hours of video are uploaded every minute and 5 billion videos are watched every day [Aslam, 2020]. On Facebook, 500 million viewers watch 100 million hours of video content daily, and 65% of the video views come from mobile users [99firms, 2019]. This volume of visual information has created the need for the development of efficient analysis and understanding methods supporting security applications, such as video search, retrieval and summarization. Understanding, in this context, refers to automatic annotation of the video with semantic concepts, e.g., objects, locations, animals, events, etc. For instance, the YouTube-8m (YT8M) video dataset [Lee et al., 2018] is annotated using a vocabulary of 3862 concepts, such as, parade, university, hunting, weapon, mountain pass, police officer, and other. In this section, we will bring expertise from other application domains, and extend top-performing deep learning techniques in order to support concept-based image/video search within the MIRROR system.

2.1.2 State of the art

In the last decade, there has been significant progress in video classification, further propelled by the introduction of large-scale benchmarking activities such as ImageNet [Russakovsky et al., 2015], TRECVID [Awad et al., 2019] and YouTube-8M [Lee et al., 2018]. Early approaches, typically employing hand-crafted local features (SIFT, HOG, SURF, etc.) combined with a feature encoding approach (e.g. VLAD, Fisher Vectors), feature pooling, and a shallow learning technique (SVM, Random Forest, etc.) were proven reasonably effective in these challenging problems [Arestis-Chartampilas et al., 2015]. The use of improved dense trajectories (IDT), exploiting more effectively longer temporal dependencies offered a further performance gain [Wang and Schmid, 2013].

Inspired by the ground-breaking results of AlexNet [Krizhevsky et al., 2012] in the ImageNet contest for still image recognition, several authors started investigating the application of deep convolutional neural networks (DCNN) for video annotation. For instance, in [Karpathy et al., 2014], DCNNs are used as feature extractors on static keyframes, and temporal feature pooling strategies are investigated. In [Ng et al., 2015], the work of [Karpathy et al., 2014] is extended by applying LSTMs in order to learn how to aggregate video-level features across multiple frames. In [Donahue et al., 2017], convolutional layers as in DCNNs and long-range temporal recursion implemented using LSTM layers, are combined, providing an end-to-end trainable architecture. In [Gan et al., 2015], a variant of AlexNet is employed to derive a deep feature vector for each keyframe, followed by max-pooling to achieve a feature representation at video level, and logistic regression is employed to annotate the video. In [Mettes et al., 2016], the semantic relationships of WordNet are used to reorganize ImageNet hierarchies and train a GoogleLeNet, which is then utilized as feature extractor in video keyframes, and SVMs are used for annotating the video. In [Simonyan and Zisserman, 2014], a two stream DCNN is designed with separate spatial (raw stacked frames) and temporal (dense optical flow) video processing streams, which are then combined by late fusion using softmax scores. In [Feichtenhofer et al., 2016], the above approach was extended using feature map fusion at different network depths. In [Ma et al., 2018], IDTs and DCNN features encoded using Fisher vector quantization and VLAD, respectively, are used to train a number of conventional classifiers (multiple kernel learning, SVM) for video annotation, additionally exploiting semantic correlation among different concepts. In [Pittaras et al., 2017], different fine-tuning strategies are investigated for video concept annotation. In [Markatopoulou et al., 2016b], [Markatopoulou et al., 2016a], [Markatopoulou et al., 2019], new methods for exploiting implicit and explicit concept relations during DCNN training are developed for improving the video annotation performance.

During the last few years, many authors have also explored the possibility of using 3D CNNs in order to directly extract spatiotemporal features from videos. In [Ji et al., 2013], a 3D convolution operator is applied separately for each color channel in both space and time, and different color channels are combined in subsequent layers of the network in order to learn spatiotemporal video features. In [Tran et al., 2015], a 3D DCNN with eleven layers for learning over 16-frame video clips was proposed, exhibiting promising performance. In [Hara et al., 2017], ResNets were extended using 3D convolutional layers in order to better exploit the spatiotemporal video information. In [Varol et al., 2018], the idea of learning much longer video sequences using appropriate 3D convolution operators is investigated in order to allow capturing the full temporal context in videos. In [Qiu et al., 2017], a Pseudo-3D Residual Net (P3D ResNet) is proposed which is more efficient to train. In [Carreira and Zisserman, 2017], inflation of 2D kernels pre-trained on ImageNet into 3D is used in order to avoid the overfitting problem of 3D artchitectures. Based on the above work, in [Hara et al., 2018], a ResNet-based 3D DCNN trained on the Kinetics dataset is fine-tuned on other visual datasets (UCF-101, HMDB-51) achieving improved performance. In [Hussein et al., 2019], spatiotemporal convolutions are decomposed into multi-scale temporal convolutions, which are better suited for modeling

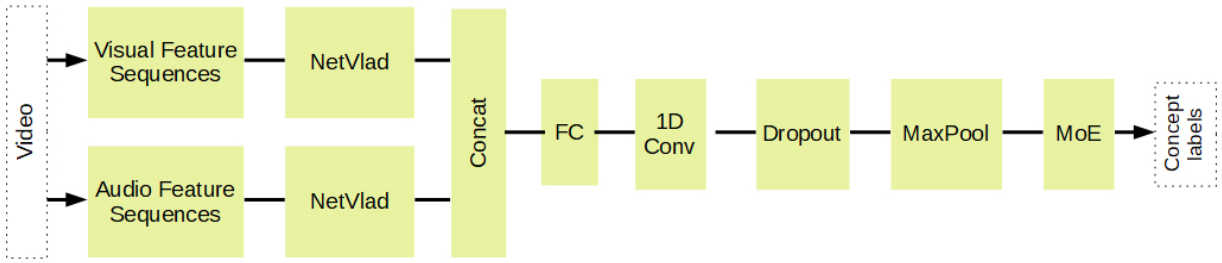

Figure 2: Overview of the proposed model architecture (Concat, FC, 1D Conv, Dropout, MaxPool and MoE denote Concatenation, Fully Connected, 1D Convolutional, Dropout, Max Pooling and Mixture of Experts Layers, respectively).

long-range events and reduce the complexity of 3D convolutions.

Despite their good performance in some application domains, 3D DCNNs have been shown to be prone to overfitting and are very expensive to train. To this end, learnable pooling architectures are recently getting increasing attention. The authors in [Girdhar et al., 2017] introduced a discriminant learning encoding, called ActionVLAD, and extend the two stream network introduced in [Feichtenhofer et al., 2016] in order to derive more discriminant spatial descriptors and aggregate frame-level features across time. In [Miech et al., 2017], the architecture introduced in [Girdhar et al., 2017] is used to model the audio and visual streams of videos, achieving the best performance in the YT8M large-scale video understanding challenge. In [Hussein et al., 2020], ActionVlad pooling [Girdhar et al., 2017] is combined with a permutation invariant convolution (PIC) layer in order to model the temporal structure of long-range activities more effectively.

2.1.3 MIRROR approach

The starting point of our work in MIRROR is our own video detection method presented in [Gkalelis and Mezaris, 2020], which has achieved very good performance in video classification despite the fact that only a simple mean pooling of frame-level feature vectors was used for video representation. In MIRROR, we further extend this approach by adopting some of the best practices introduced in the top-performing architecture of [Miech et al., 2017]. Specifically, a NetVlad layer is added in the input of the network in order to learn a more discriminant video encoding, and a Mixture of Experts layer is used to remove nonlinearities of the learned features in the output layer. The overall architecture is shown in Figure 2, and described in more detail in the following.

Let $\mathcal{D}$ be an annotated training set of N vector sequences $\mathbf{X}_n = [\mathbf{x}_{n,1}, \ldots, \mathbf{x}_{n,T}] \in \mathbb{R}^{F \times T}$

$$\mathcal{D} = \{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_N, \mathbf{y}_N)\}, \tag{2.1}$$

where, $\mathbf{y}_n = [y_{n,1}, \ldots, y_{n,C}]^T \in \mathbb{R}^C$ is the class indicator vector of the nth sequence (i.e. its ith element $y_{n,i}$ is one if $\mathbf{X}_n$ belongs to class i and zero otherwise), C is the total number of classes, $\mathbf{x}_{n,t} = [x_{n,t,1}, \ldots, x_{n,t,F}]^T \in \mathbb{R}^F$ is the tth vector of the ith sequence, $x_{n,t,j}$ is the jth element of the $\mathbf{x}_{n,t}$, F is the input space dimensionality and T is the sequence length (without loss of generality it is assumed that all sequences have the same length). More specifically, in the context of MIRROR, $\mathbf{X}_n$ represents the nth video in a training database of N videos in total. Similarly, the feature vector $\mathbf{x}_{n,t}$ corresponds to the tth frame of the nth video, and is derived using an appropriate feature descriptor (e.g. SURF, HOG or the last layer of an entire pretrained DCNN) applied in the frame.

For the automatic classification of unlabeled vector sequences the one-dimensional deep convolutional neural network (1D DCNN) architecture presented in [Gkalelis and Mezaris, 2020] is utilized. More specifically, in order to allow the effective exploitation of the discriminant information in temporal sequences the 1D DCNN is extended using a NetVlad input layer [Girdhar et al., 2017]. In this way, the nth sequence in the output of the NetVlad layer is represented as a KF-dimensional vector $\mathbf{v}_n = [v_{n,1,1}, \dots v_{n,F,K}]^T$,

$$v_{n,j,k} = \sum_{t=1}^{T} \frac{\exp(-\alpha||\mathbf{x}_{n,t} - \mathbf{m}_k||^2)}{\sum_{k'=1}^{K} \exp(-\alpha||\mathbf{x}_{n,t} - \mathbf{m}_{k'}||^2)}(x_{n,t,j} - m_{k,j}), \tag{2.2}$$

$$= \sum_{t=1}^{T} \frac{\exp(\mathbf{w}_k^T \mathbf{x}_{n,t} + \mathbf{b}_k)}{\sum_{k'=1}^{K} \exp(\mathbf{w}_{k'}^T \mathbf{x}_{n,t} + \mathbf{b}_{k'})}(x_{n,t,j} - m_{k,j}), \tag{2.3}$$

where, $\mathbf{m}_k = [m_{k,1}, \dots, m_{k,F}] \in \mathbb{R}^F$ is the kth anchor vector representing the kth "word" in the NetVlad vocabulary, $m_{k,j}$ is the jth element of the kth anchor vector, $\mathbf{w}_k = 2\alpha\mathbf{m}_k$, $\mathbf{b}_k = -\alpha||\mathbf{m}_k||^2$ are the weight vector and bias of the NetVlad layer, and α is a tunable hyper-parameter.

In order to process both the visual and audio feature sequences we use a two stream architecture followed by a concatenation and a fully-connected (FC) dense layer to combine the feature maps coming from the two streams (e.g. similar to [Miech et al., 2017], [Girdhar et al., 2017]). Additionally, the sigmoid output layer of our architecture in [Gkalelis and Mezaris, 2020] is replaced by a mixture of experts (MoE) layer which has shown very good performance in several supervised learning applications [Wang et al., 2018b]. That is, using the MoE layer the output of the network $\hat{y}_{n,i}$ with respect to class i and sequence n is computed as

$$\hat{y}_{n,i} = \sum_{p=1}^{P} g_{i,p,n} e_{i,p,n}, \tag{2.4}$$

where, $e_{i,p,n}$ is the output of the pth expert and $g_{i,p,n}$ is the weight assigned from the gating function to the above expert.

The overall proposed DCNN architecture is depicted in Figure 2. It consists of a NetVlad, concatenation, hidden fully-connected (FC) dense, 1D convolutional, dropout, max-pooling and MoE layer, and can be trained end-to-end using standard deep learning optimization procedures.

2.1.4 Experiments, datasets, results and future work

For the evaluation of the proposed DCNN architecture we utilize the second version of the YouTube-8M (YT8M) dataset [Lee et al., 2018]. This is the largest publicly available multilabel video dataset consisting of 6.1 million videos, 3862 classes and 3 labels per video on average. Visual and audio feature vector sequences are extracted at frame-level (1 frame per second). Visual features are obtained from Google's Inception-v3 model pretrained on Imagenet [Russakovsky et al., 2015], while audio features are extracted using a VGG-inspired model trained for audio classification as described in [Hershey et al., 2017]. Principal component analysis (PCA) is applied to both visual and audio features to further reduce their dimension to 1024 and 128 respectively. The dataset is divided to a training, validation and testing partition, consisting of 3888919, 1112356 and 1133323 videos, respectively. The data are stored in Tensorflow's tfrecord file format (3844 shards for each data partition), which offers very efficient import and preprocessing functionalities for large-scale datasets. We only use the training and validation partitions because the ground truth labels of the testing partition are not publicly available.

Table 1: Evaluation results on the YT8M dataset.

	GAP@20	*mAP*	*PERR*	*Hit@1*
MP+SG	77%	40.1%	71.8%	82.3%
MP+DCNN+SG	80.7%	45.6%	75.4%	85.2%
SDCNN [Gkalelis and Mezaris, 2020]	82.2%	47.9%	75.9%	85.7%
NVLAD+SG	84.1%	49.4%	78.8%	87.9%
NVLAD+MoE	84.9%	50.4%	**79.3%**	**88.2%**
NVLAD+DCNN+MoE (MIRROR approach)	**85.1%**	**50.5%**	**79.3%**	**88.2%**

In order to utilize the discriminant information of both the visual and audio modalities a two-stream architecture is used as described in the section above and shown in Figure 2. The number of anchor vectors in the NetVlad layer is set to $K = 256$ and $K = 128$ for the visual and audio stream, respectively. On the other hand, the configuration of the rest of the layers in both architectures are the same for both streams as described in the following. The size of the hidden FC linear layer is set to $H = 1024$ for both streams. The convolutional layer consists of 16 1D filters with receptive field of size 3 and stride 1, rectification (ReLU) nonlinearity, and zero padding is applied in order to preserve the spatial size of the input signals. A keep-rate of 0.7 is applied on the dropout layer and the max-pooling layer uses a a window of size 2 and stride 2. For the MoE layer we use 3 experts for each class (with one of them representing the rest-of-the-world expert category) the sigmoid and softmax nonlinearity is used for the gaiting and expert function, respectively, and the weight regularization term for both the gaiting and expert function is set to 10^{-6}.

The proposed network architecture (denoted hereafter as NVLAD+DCNN+MoE) is compared against several top-performing approaches, namely, mean pooling (MP) of the frame-level visual and audio features and the use of a sigmoid output layer (MP+SG), MP combined with a rather simple 1D DCNN with SG output layer (MP+DCNN+SG), our baseline approach (SDCNN) described in [Gkalelis and Mezaris, 2020], the NetVlad layer combined directly with a SG (NVLAD+SG) or a MoE output layer (NVLAD+MoE). All the networks are trained using cross entropy (CE) loss, Adam optimizer with an exponential learning rate schedule, 3 epochs and minibatches of 160 videos. The official evaluation metrics of YT8M challenge are used for measuring the performance of each method, namely, Hit@1, precision at equal recall rate (PERR), mean average precision (mAP), and global average precision at 20 (GAP@20). The latter is the primary evaluation metric of the YT8M challenge for ranking the different participating teams. It is calculated by first sorting the predictions for each video according to the confidence score and then computing the average precision (AP) along all predictions and all videos:

$$AP = \sum_{q=1}^{Q} \pi(q)\Delta r(q), \tag{2.5}$$

where, $Q = 20 \times$ total number of videos, and $\pi(q)$, $\Delta r(q)$ are the precision and recall given the first q predictions.

The evaluation results along the different methods and evaluation metrics are shown in Table 1. Moreover, Figure 3 depicts four example keyframes and detection scores from five test videos that were classified correctly using the proposed approach. From the obtained results we conclude the following: i) The NetVlad+SG outperforms SDCNN by almost 2%. This is due to the use of the NetVlad layer, which by learning a discriminant encoding mechanism, provides a clear advantage over the MP method for sequence representation. ii) A further 0.8% performance gain is achieved by NetVlad+MoE architecture over NetVlad+SG.

Figure 3: Four example keyframes from five test videos that were classified correctly using the proposed approach along with the detection scores.

Thereofore, we observe that the MoE layer has also a beneficial effect in the classification performance, where it is used to replace the SG layer. iii) Finally, using the proposed architecture (NetVlad+DCNN+MoE) we achieve an additional 0.2% absolute performance gain. Although this is a relatively small gain, we consider it a quite significant improvement considering that the NVLAD+MoE approach is already a very strong network that has achieved state-of-the-art results in this domain.

As a future work direction we will consider the use of semi-supervised learning techniques in order to use effectively unlabeled video sequences. To this end, we have started the investigation of appropriate techniques extending the NVLAD+DCNN+MoE architecture, so that we can exploit the discriminant information of the unlabeled video sequences in the test partition of the YT8M dataset.

2.2 Video captioning

2.2.1 Problem statement

Video Captioning is the task of describing the content of a video using natural language. Such a task is important because it helps to search through large video databases by using queries in natural language. Automatic video captioning is also useful for finding videos that relate to each other, or, complementing the concept-based annotation discussed in the previous section, detecting videos where specific concepts appear (e.g. armed robbery).

Automatic Video Captioning is a hard problem. Current methods try to make use of various modalities like optical flow and audio, or use ”attributes” inside the video such as by detecting objects, sounds and actions. The major problem is that analysing the video files require vast amounts of memory. Many approaches use attention mechanisms to select specific frames to work with, while others use feature extractors and work

directly on the features instead of the raw video frames. Still these methods had limited results. Newer approaches try to model the hierarchy inside the video and seem to give some promising results combined with attribute detection.

2.2.2 State of the art

The first attempt to the problem of Video Captioning was made using classic Encoder-Decoder methods like S2VT by [Venugopalan et al., 2015] where the authors approached the problem as a sequence-to-sequence. One way to improve the results was to use multiple modalities. The topic diversity of open-domain videos maes the task of captioning extremely challenging. [Chen et al., 2017] tried to solve this problem by fine-tuning the accuracy through the detection of learnable concepts which they call topics. The mined topics are then used as the teacher to train a student topic prediction network. The method proposed by [Jin et al., 2016] follows the encoder-decoder architecture but makes use of five modalities, video, image, aural, speech, and video metadata, and achieves good results. This method still holds the first place in the MSR-VTT leaderboard.

A novel approach to the video captioning problem is through the usage of semantic attributes. [Sun et al., 2019] approach video captioning in a multimodal way making use of image, video, optical flow and audio modalities. Specifically they detect objects in the frames and they use optical flow to detect actions (like playing, running, etc). Through the concatenation of those features they achieve state of the art results. In their work, [Zhang and Peng, 2019] detect objects and try to learn the dynamics among the detected objects by inserting them in a bidirectional temporal graph.

2.2.3 MIRROR approach

The approach we used as a starting point for our experiments in MIRROR is the HACA model presented in [Wang et al., 2018a]. The HACA model is an encoder-decoder framework comprising multiple recurrent neural networks. In the paper two modalities were used, the visual and the audio, but the model can easily extend to use more modalities, such as optical flow. The model has a hierarchical attentive encoder for each input modality that learns and outputs both the local and global representation of the modality. In the decoding phase two attentive decoders are implemented, a local decoder and a global decoder. The global decoder tries to align the global context of all the modalities while the local decoder learns a local cross-modal fusion context, combines it with the output of the global decoder and predicts the next word.

2.2.4 Experiments, datasets, results and future work

There are various datasets that are being used for Video Captioning benchmarks. The most common ones are MSR-VTT [Xu et al., 2016] which consists of 10000 videos taken fromYouTube, MSVD [Chen and Dolan, 2011] which consists of 1970 videos taken also from YouTube and MPII-MD [Rohrbach et al., 2015] which consists of 68337 videos taken from Movies.

The most common dataset on Video Captioning is MSR-VTT. In the first version that came out in 2016 it has 6513 videos for training, 497 videos for validation and 2990 videos for testing. In the 2nd version that came out in 2017 all the 10000 videos were used for training and another 3000 videos were added for testing purposes. What makes this dataset ideal for video captioning is the fact that every video has 20 captions.

We executed various experiments to optimize and extend the HACA baseline model using the MSR-VTT dataset. First we tried increasing the sampling rate and investigated how the increased sampling in frames affected the accuracy. We see in the following table (Table 2) that there was a slight improvement according to some of the employed evaluation metrics.

Table 2: Comparison of results when using a 3fps and a 6fps sampling rate.

Method	Bleu4	Meteor	Rouge	Cider
3fps	**0.311**	0.35	**0.612**	0.24
6fps	**0.311**	**0.352**	0.608	**0.251**

A second set of experiments was by implementing optical flow as an extra modality. Unfortunately, as can be seen in the following Table 3, adding optical flow slightly decreases the performance.

Table 3: Results with and without using optical flow.

Method	Bleu4	Meteor	Rouge	Cider
default	**0.311**	**0.35**	**0.612**	**0.24**
with optical flow	0.298	0.349	0.6	0.237

In a third set of experiments we tried to introduce global features early in the model to help the training converge faster. The global features were the average value of all the frame features, and they were concatenated to the feature vector of each frame. However this had no impact on the results (Table 4).

Table 4: Early fusion of global video features.

Method	Bleu4	Meteor	Rouge	Cider
default	**0.311**	**0.35**	**0.612**	**0.24**
avg pooling	0.306	**0.35**	0.607	0.239

Finally in a fourth set of experiments we tried ignoring the audio modality for the videos where there was no audio. In the previous experiments we used arrays of zeros to represent the missing audio. The results are shown in the following Table 5. We can see that this minor tweak of the method introduces a slight improvement.

Table 5: Ignoring missing audio

Method	Bleu4	Meteor	Rouge	Cider
default	0.311	0.35	**0.612**	0.24
ignore audio	**0.315**	**0.353**	0.61	**0.251**

The HACA method served in this early set of experiments as a MIRROR baseline approach to work on, and produce a working implementation that can be used in MIRROR. Based on our experiments we optimized the initial implementation that has been integrated in the MIRROR system (as discussed in Section 2.4 below) to adapt the final configuration (”ignore audio”) reported in Table 5. As next steps, inspired by the work of [Yao et al., 2019], where they build a hierarchy of the image by detecting regions and items in an image, we intent to follow a similar approach in video. Furthermore we intent to expand the method

by making use of motion dynamics by incorporating to the aforementioned method 3d convolutional and optical flow networks.

In Figure 4 we can see some example frames of videos of the MSR-VTT dataset and the automatically-generated captions that the MIRROR implementation produced.

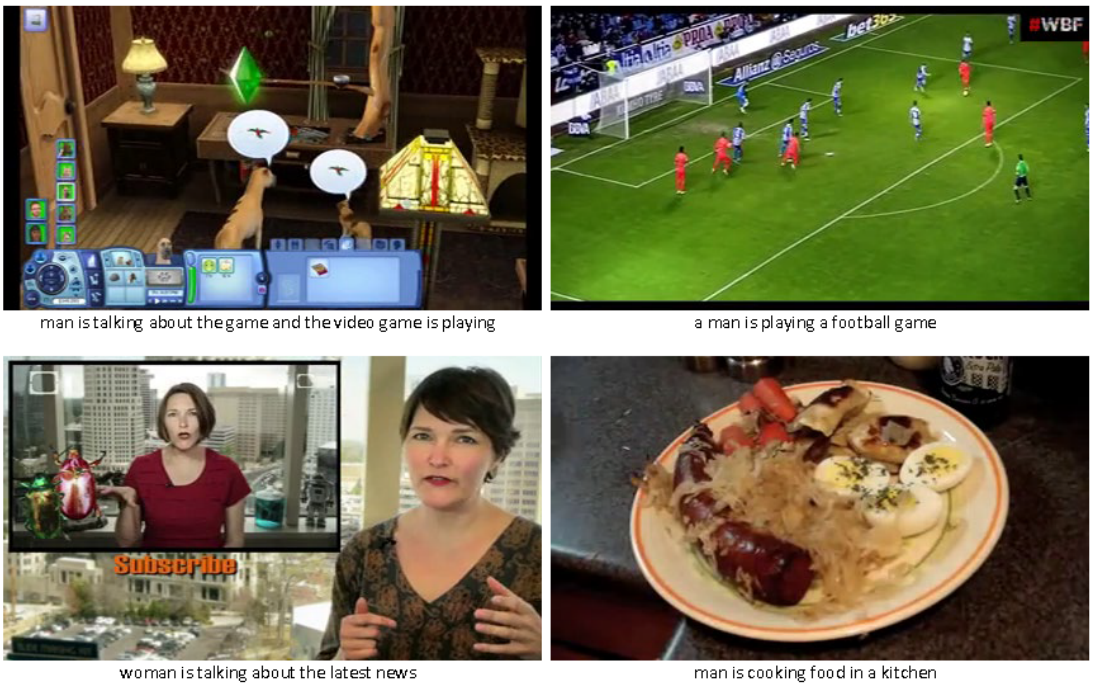

Figure 4: Example results of video captioning. Generated captions for different video shots.

2.3 Migration-related semantic concept detection

2.3.1 Problem statement

The goal of the Migration-Related Semantic Concept (MRSC) detection method is to retrieve video shots that are related to a given MRSC. The main challenge of this task is to associate MRSCs, which in general are abstract textual concepts, e.g. *ethnic identity*, *law enforcement* etc., with visual content, without using any training visual exemplars. Typically, a concept annotation and retrieval method (as in Section 2.1) uses image/video exemplars to train pre-defined concept detectors. However, for the needs of the MIRROR project, it is extremely difficult to find and annotate large numbers of exemplar images/videos that are associated with the MRSCs in order to train detectors for each MRSC. Moreover, the MIRROR system should be able to handle new content (and new MRSCs) without any time-consuming procedures such as manual annotation and training. For these reasons, we adopt a state-of-the-art approach for Ad-hoc video search in order to develop a method that can directly transform visual and textual context into a common feature space, in which a straightforward comparison is feasible.

2.3.2 State of the art

We address the MRSC detection problem as a special type of zero-shot learning (ZSL). Typically the ZSL refers to the problem in which the classes used to train a classifier and the classes used for evaluation purposes, do not overlap. For example, a ZSL method for object recognition aims to recognize an object that occurs in an image or video that is not included in the training set. Since the classic object recognition challenges that are based on using abundant training exemplars, such as the ImageNet classification challenge, have achieved higher than human performance [Xian et al., 2017], the more challenging problem of detecting unknown classes has attracted researchers' attention. As the definition of the ZSL task implies, supervised learning methods are not useful. Early approaches [Lampert et al., 2014] [Norouzi et al., 2013] deal with the problem as a mixture of sub-problems and use attribute-based methods to detect unseen classes. First, given as input an image, the attributes of the image are identified and then they search for a class that contains the found attributes. Recent approaches such as [Akata et al., 2016] learn to directly map the image feature space into the semantic feature space and then the most related classes are ranked.

Similar to ZSL, cross-modal retrieval aims to bridge the gap between different feature spaces from different modalities. For example, given an image, a cross-modal retrieval system aims to find the most related image caption or audio and vice versa [Wang et al., 2017] [Zhen et al., 2019]. As the migration-related semantic concept detection problem aims to annotate images or video shots with MRSCs for the purpose of retrieval, we focus on the problem of the Ad-hoc Video Search (AVS), which is a special type of cross-modal retrieval problem in which video shots must be retrieved when the query is a complex textual sentence.

The first attempts on the AVS problem have relied on large sets of pretrained visual concept detectors, and NLP techniques for query analysis to find relevant visual concepts in the query. In [Markatopoulou et al., 2017], the association between visual concepts and the textual queries was reached by using complex NLP rules and a vast set of pre-trained deep neural networks for video annotation. More recently, the problem has been addressed by using deep neural networks to transform both the textual queries and the visual content in a new common space [Habibian et al., 2017]. The dual encoding network proposed in [Dong et al., 2019] uses multiple levels of encodings to transform videos and queries into a common dense representation using the improved loss function of [Faghri et al., 2018]. An extension of the above is presented in [Galanopoulos and Mezaris, 2020] were SotA results were achieved using a richer representation and attention-based layers of encodings.

2.3.3 MIRROR approach

As an early approach for the MRSC detection problem, we associate automatically each MRSC with a set of the most related visual concepts (VCs), e.g. such as those detected with the method of Section 2.1. So, to retrieve video shots for a specific MSRC the pre-trained detectors of the associated visual concepts would be used. For this, a word embeddings method is utilized to represent a free-text sentence as a low-dimensional vector. Every MRSC and VC are vectorized using a pretrained language representation model i.e. BERT [Devlin et al., 2018]. BERT generates multiple, contextual, bidirectional word representations that are used to find the most related VCs to an MRSC by directly comparing every pair of MRSC-VC vectors. However, this approach is suffering from some disadvantages. In general, MRSCs are abstract concepts and it is extremely difficult to be associated with more specific concepts like VCs. Moreover, in the case of a new MRSC is ingested to the system, the procedure to update the framework is time-consuming, because new VCs must be associated with the new MRSC, and then positive video samples must be found and finally, the new VC detectors must be trained.

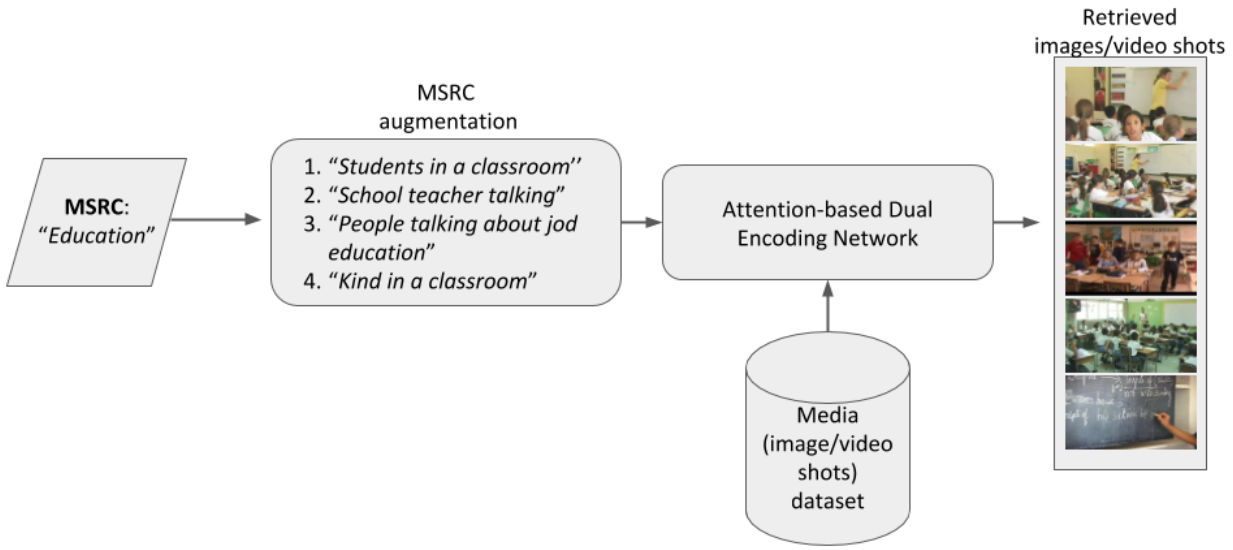

Figure 5: An overview of the MIRROR MRSC detection method.

To overcome all these limitations, we addressed the MRSC detection problem by adapting a previously developed state-of-the-art method for the AVS problem. This method aims to retrieve unlabeled video shots when the input is a textual query. For this, we adjust the attention-based dual encoding network presented in [Galanopoulos and Mezaris, 2020]. The network utilizes two similar modules, consisting of multiple levels of encoding, for the visual and textual content respectively, along with a text-based attention component for more efficient textual representation. More specifically, the network translates a media item (e.g., an entire video or a video shot) **V** and a textual item (e.g. a caption or a text query) **S** into a new shared feature space $\Phi(\cdot)$, resulting in two new representations $\Phi(\mathbf{V})$ and $\Phi(\mathbf{S})$ that are directly comparable.

The overall idea of this approach is to train a deep network by using video-caption pairs and then to use it as a video retrieval system by inputting MRSCs, in order to retrieve the most related video shots w.r.t. the input MRSC. As the MRSCs typically consist of one or two words, we choose to enrich them in order to be more manageable by the video retrieval system. Each MRSC is manually augmented with a small set of complex sentences that describe it. For instance, for the MRSC *education*, sentences like *students in a classroom attend a lecture* are generated. This approach is closely related to the training procedure of our network. The MRSC with its descriptions and the video shots from the target dataset are used as input to our system. They are encoded into the common feature space and for every MRSC a ranked list with the most related media items within the given image/video dataset is generated. An overview of the proposed method is illustrated in Fig 5.

2.3.4 Experiments, datasets, results and future work

To train our network we used the combination of two large-scale video datasets: MSR-VTT [Xu et al., 2016] and TGIF [Li et al., 2016]. As initial frame representations, we use a ResNet-152 (trained on the ImageNet-11k dataset). Also, two different word embeddings are utilized: i) the Word2Vec model [Mikolov et al.,

2013] trained on the English tags of 30K Flickr images, provided by [Dong et al., 2018]; and, ii) the pre-trained language representation BERT [Devlin et al., 2018], trained on Wikipedia content. To evaluate the performance of our network for the purposed of the MRSCs detection, since there is no available dataset for this, we use the evaluation dataset of the TRECVID Semantic Indexing task (SIN) task for the years of 2013 and 2015. The goal is to retrieve the most related video shots by inputting the names of a set of visual concepts. These concepts take over the position of the MRSC for evaluation purposes. Good performance on these datasets will document the merit of the following approach for the needs of MIRROR. The mean extended inferred average precision (MXinfAP) is used as an evaluation measure.

Also, we compare our approach with conventional concept retrieval methods which use predefined sets of visual concepts, positive exemplars for every concept, and were trained specifically for these. The goal of these comparisons is to highlight the performance of our approach even if no concept specific training videos are used, in contrast to the supervised learning methods. These comparisons validate the capability of this method to meet the special needs of the MRSC detection task.

Table 6 presents the results on the SIN 2013 and SIN 2015 dataset for the detection of 38 and 30 different concepts, respectively. The results of Table 6 show that the utilization of additional information for every concept (the "augmentations" that were mentioned in the previous subsections, i.e. additional sentences that describe the concept, besides its label) leads to significantly improved performance. Typical examples of great performance improvements are the concept *"Telephones"* in both datasets, and *"Bicycling"* and *"Demonstration Or Protest"* in the SIN 2015 dataset. The *"Telephones"* concept was described as *"speaking on a telephone"* and *"talking on a telephone"* and achieved XinfAP improvement from 0.0 to 0.3151 and from 0.0 to 0.308 in the SIN 2013 and SIN 2015 datasets, respectively. Similarly *"Bicycling"*, which was described as *"a man riding a bike"*, *"people riding bicycles"* and *"a woman on a bike"* achieved 0.373 XinfAP, compared to 0.0569 when only the word "bicycling" was used.

In order to highlight the performance of our approach, we compare our results with two different methods that are designed to solve the SIN task, using training video samples. For the SIN 2013 dataset we compared with the work which was presented in [Markatopoulou et al., 2016a] and with the CERTH participation to the TRECVID SIN task in 2013 [Markatopoulou et al., 2013], while in the SIN 2015 dataset we present a comparison with the CERTH participation to the TRECVID SIN task in 2015 [Markatopoulou et al., 2015]. Table 7 shows that our approach is very competitive to these methods, even though they are designed specifically for the SIN task. It is clear that our MRSC detection approach does not achieve SotA performance on the SIN task, however, these results are strong evidence that our approach is suitable for the MRSCs detection problem where, as opposed to the TRECVID SIN task used for this comparison no training data (annotated visual exemplars) are available for the MRSCs.

As mentioned before there is no MRSC-specific dataset to evaluate the performance of our approach in a numerical way and for this reason, we presented numerical results on the TRECVID SIN datasets. However, to further validate that our approach is suitable for the needs of MIRROR, we present some visual examples of the retrieved video shots when we use the MIRROR MRSCs as input to our method. In Fig.6 the top-5 retrieved shots are presented for a portion of the available MRSCs. It is quite clear our approach succeeds to find videos shots that are related to the MRSCs.

So far, the MRSCs detection approach achieved adequate results, but there is room for improvements. Improved visual features for more efficient training and effective retrieval along with better textual descriptions will be examined. Also, the automatic generation of the textual description will be evaluated for a fully automatic pipeline. Finally, the efficiency of improved ranking list fusion methods will be examined for efficient combination of different variations of the MRSCs detection method.

Table 6: Results of video shot retrieval in the SIN 2013 and SIN 2015 datasets for 38 and 30 visual concepts, respectively in terms of XinfAP. Two different setups are presented. The ”Concept name“ column presents the results when we use as textual input only the concept label, while the ”Concept name + descriptions“ column stands for setup in which the concept label is augmented with short sentences further describing the concept.

SIN 2013 dataset			SIN 2015 dataset		
	Concept name	Concept name + descriptions		Concept name	Concept name + descriptions
1003 Airplane	0.1928	0.2789	1003 Airplane	0.3254	0.5055
1005 Anchorperson	0.0128	0.0646	1005 Anchorperson	0.0067	0.0145
1006 Animal	0.0253	0.1748	1009 Basketball	0.0134	0.1814
1010 Beach	0.4648	0.515	1013 Bicycling	0.0569	0.373
1015 Boat Ship	0.3653	0.4443	1015 Boat Ship	0.4804	0.5998
1016 Boy	0.0601	0.1279	1017 Bridges	0.085	0.1615
1017 Bridges	0.0268	0.0688	1019 Bus	0.1215	0.1382
1019 Bus	0.0657	0.112	1022 Car Racing	0	0.0647
1025 Chair	0.0309	0.1207	1027 Cheering	0.0004	0.0687
1031 Computers	0.112	0.2982	1031 Computers	0.148	0.362
1038 Dancing	0.0242	0.1503	1038 Dancing	0.0002	0.1239
1049 Explosion Fire	0.1884	0.2582	1041 Demonstration Or Protest	0	0.2574
1052 Female Human Face Closeup	0.1017	0.1459	1049 Explosion Fire	0.104	0.1739
1053 Flowers	0.1035	0.1661	1056 Government Leader	0.0003	0.1677
1054 Girl	0.0388	0.1271	1071 Instrumental Musician	0.0002	0.3458
1056 Government Leader	0	0.2767	1072 Kitchen	0.0805	0.34
1059 Hand	0.0904	0.1025	1080 Motorcycle	0.1303	0.236
1071 Instrumental Musician	0.0031	0.3305	1085 Office	0.0546	0.2425
1072 Kitchen	0.0745	0.1537	1086 Old People	0.0473	0.1993
1080 Motorcycle	0.2042	0.2581	1095 Press Conference	0.0001	0.0219
1083 News Studio	0.0206	0.0609	1100 Running	0.0008	0.0178
1086 Old People	0.0854	0.2108	1117 Telephones	0	0.3088
1089 People Marching	0	0.0626	1120 Throwing	0.0001	0.0485
1100 Running	0.0059	0.1494	1261 Flags	0.0685	0.156
1105 Singing	0.0008	0.1057	1297 Hill	0.0319	0.0675
1107 Sitting Down	0.0001	0.0084	1321 Lakes	0.0577	0.2033
1117 Telephones	0	0.3151	1392 Quadruped	0.0017	0.2311
1120 Throwing	0	0.125	1440 Soldiers	0.2436	0.3709
1163 Baby	0.2991	0.4707	1454 Studio With Anchorperson	0.0021	0.0393
1227 Door Opening	0.0177	0.0377	1478 Traffic	0.1372	0.2046
1254 Fields	0.0192	0.1578			
1261 Flags	0.1274	0.2687			
1267 Forest	0.1026	0.1939			
1274 George Bush	0	0.44			
1342 Military Airplane	0.0001	0.1062			
1392 Quadruped	0.0214	0.2928			
1431 Skating	0.2684	0.424			
1454 Studio With Anchorperson	0.0047	0.0419			
Mean XinfAP	0.0831	**0.2012**		0.0733	**0.2075**

Table 7: Comparison with SIN Task-specific methods in the SIN 2013 and SIN 2015 datasets, in terms of XinfAP.

	SIN 2013	SIN 2015
MIRROR approach (no training exemplars)	0.2012	0.2075
[Markatopoulou et al., 2016a] (using annotated exemplars for training)	0.2504	-
[Markatopoulou et al., 2013] (using annotated exemplars for training)	0.1580	-
[Markatopoulou et al., 2015] (using annotated exemplars for training)	-	0.263

Figure 6: Example results, for 5 MRSCs (shown on the top of the figure). The top-5 retrieved shots for each MRSC are shown.

2.4 Service implementation and APIs

For the purpose of MIRROR, CERTH has setup a dedicated server that hosts all the multimedia analysis methods that comprise the Visual Media Annotation and Sentiment component (MAS). The results of the different analysis methods of the aforementioned tasks are available to the main MIRROR framework as a REST. This approach significantly reduces the processing burden of the main server hosting the MIRROR framework, as the multimedia analysis methods are relatively heavy on computer resources and require specific GPU hardware.

The initial version of the MAS supports the analysis of images and videos with the methods that have been described in this section, and produces the respective annotations for each method: (a) visual concepts, (b) captions and (c) MRSCs. The images or videos must reside in a downloadable URL location or one of the popular social media or video hosting platforms that are supported [1]. Image analysis can be performed on single images or image collections/groups. In both cases, annotations are produced for each image individually. For videos, the analysis is performed on a temporal segment level. Three levels of temporal fragmentation are supported: scenes (i.e. semantically and temporally coherent segments that correspond to the story-telling parts of the video), shots (i.e. sequences of frames captured uninterruptedly by a single camera) and sub-shots (i.e. sub-parts of a shot with visually discrete content; useful when analyzing single-shot videos). Following fragmentation, the MAS performs the visual analysis for each fragment. The supported visual concept pool at this point is the Youtube-8M [2], comprising over 3000 visual concepts.

MAS follows the Representational State Transfer (REST) web service architecture. Its functionalities are accessed through its REST API, which is a set of HTTP calls. Since multimedia processing is time consuming, the MAS functions in an asynchronous manner. From the client side, the asynchronous workflow follows these basic steps:

1. Issue request to start the analysis of a multimedia item
2. Inquire about the status of the analysis
3. When the analysis is done, retrieve the results

The base URL of the MAS is http://mirror.iti.gr. A request to initiate the processing of a media item can be issued with POST http://mirror.iti.gr/image-annotation or POST http://mirror.iti.gr/video-annotation. The first call will request the annotation of an image collection while the latter will request the annotation of a video. The URL of the media item and other parameters are passed in the body of the POST request in JSON format. Some of them are:

- **images_url** The URL of the image collection.
- **video_url** The URL of the video to be processed.
- **user_key** A key used for authentication/security reasons.
- **concept_list** Defines the concept pool to be used for the annotation (currently only Youtube-8M is supported and is default).

To clarify the functionality of the MAS, we provide an example of the workflow required to analyse a video and retrieve the results. The process follows for analyzing an image collection is very similar. One can

[1] Youtube, DailyMotion, Facebook, Twitter and Vimeo are among the supported social media/video hosting platforms
[2] https://research.google.com/youtube8m/explore.html

request the fragmentation and annotation of a video hosed on http://host.eu/video.mp4 by issuing the following request:

```
POST http://mirror.iti.gr/video-annotation
{
  "video_url": "http://host.eu/video.mp4",
  "user_key": "0123456789"
}
```

The reply to the previous call will be a JSON file containing a message and a item_id that identifies the media item and will be later used to inquire about the status of the analysis and to retrieve the results.

```
{
  "message": "The REST call has been received. Please check the status of
              the analysis via the appropriate REST call",
  "item_id": "f75bnfg2nfncgtn347572839"
}
```

The status of the analysis can be inspected with the following call:

```
GET http://mirror.iti.gr/status/<item_id>
```

where <item_id> is the item_id previously retrieved from the POST request. The status call returns a small message describing the state of the analysis of the specific video and whether it has finished or not. Some of the status messages are:

- VIDEO_WAITING_IN_QUEUE
- VIDEO_DOWNLOAD_STARTED
- VIDEO_SEGMENTATION_ANNOTATION_STARTED
- VIDEO_SEGMENTATION_ANNOTATION_COMPLETED

If the status message informs that the video analysis has finished successfully, then the analysis results can be retrieved in a JSON format by issuing the following GET request:

```
GET http://mirror.iti.gr/result/<item_id>_json
```

The document returned complies with the following structure:

```
framerate
generated_at
expires_at
generated_by
version
list of scenes contained in the video
--> scene #i
 --> begin_time (in sec)
 --> end_time (in sec)
 --> list of keyframes for the scene
  --> keyframe #n (by default, n = [1,5])
   --> time (in sec)
   --> url
 --> list of shots contained in the scene
```

```
  --> shot #j
   --> begin_time (in sec)
   --> end_time (in sec)
    --> list of keyframes for the shot
     --> keyframe #p (by default, p = [1,3])
      --> time (in sec)
      --> url
   --> top concepts for the shot
    --> concept #m: confidence score
   --> top MRSCs for the shot
    --> MRSC #m: confidence score
   --> caption
   --> list of sub-shots contained in the shot
    --> sub-shot #k
     --> begin_time (in sec)
     --> end_time (in sec)
    --> list of keyframes for the sub-shot
     --> keyframe #p (by default, p = [1,3])
      --> time (in sec)
      --> url
    --> top concepts for the sub-shot
     --> concept #m: confidence score
    --> top MRSCs for the sub-shot
     --> MRSC #m: confidence score
    --> caption
   --> end of list of sub-shots contained in the shot
 --> end of list of shots contained in the scene
end of list of scenes contained in the video
```

The extracted information about the automatically defined temporal fragments (i.e. scenes, shots and sub-shots) and semantic annotations of the video is structured in a coarse-to-fine schema in the produced JSON file after the end of the analysis. In particular, the outer level of this structure provides information about the video frame-rate, the date and time when the file was created, the date and time when the file expires (i.e. it is automatically deleted by the service), the service that generated the file, the version of the service, and the scenes of the video, where each scene is described by: (a) an identifier (called scene_id), (b) its begin and end time in seconds, (c) a number (up to 5) of representative keyframes with their timestamps (i.e. the time of appearance in the video) and their URLs that make them downloadable, and (d) the group of shots that constitute the scene. The latter forms the next (mid) level of the JSON structure that provides information about the shots of the video. Each shot is described by: (a) an identifier (called shot_id), (b) its begin and end time in seconds, (c) a number (3 by default)) of representative keyframes with their timestamps (i.e. the time of appearance in the video) and their URLs that make them downloadable, (d) the top visual concepts that were detected with the highest confidence scores, (e) the top MRSCs that were detected with the highest confidence scores, (f) a caption for the shot, and (g) the group of sub-shots that compose the shot. The latter forms the next (inner) level of the JSON structure that provides information about the sub-shots of the video. Each sub-shot is described by (a) an identifier (called subshot_id), (b) its begin and end time in seconds, (c) a number (3 by default)) of representative keyframes with their timestamps (i.e. the time of appearance in the video) and their URLs that make them downloadable, (d) the top visual concepts that were detected with the highest confidence scores, (e) the top MRSCs that were detected with the highest confidence scores, and (f) a caption for the sub-shot.

The analysis results for an image collection follows a much simpler structure:

```
framerate
generated_at
expires_at
generated_by
version
list of images contained in the collection
--> image #i
 --> image_name
 --> top concepts for image
  --> concept #m: confidence score
 --> top MRSCs for the shot
  --> MRSC #m: confidence score
 --> caption
end of list of images contained in the collection
```

The outer level of the document contains the list of images contained in the collection. Each image is identified by its image_name and contains the annotation fields (concepts, MRSCs, caption) that follow the same structure as in the video.

These results files, as soon as they are generated by the MAS component described here, are ingested in the MIRROR system as foreseen in D7.1.

3 Visual sentiment analysis

The detection of misperceptions about Europe is one of the key aims of the MIRROR project, as misperceptions can lead to security threats. The media's sentiment is one of the most fundamental indicators of the perception of Europe, in the various countries of origin of potential migrants. Sentiment can be detected not only from text but also from image and video sources. In this section we focus on the problem of detecting the sentiment from images, which can also be applied on videos.

3.1 Problem statement

Images and videos are a powerful tool to convey information and sentiment. We refer to sentiment to indicate the subjective bias of an opinion, usually in terms of positive or negative. A picture of an abandoned building in a cloudy day conveys a more negative sentiment than a picture of a happy family in a sunny day. The task of visual sentiment analysis tries to solve the problem of determining the sentiment of images and videos.

The task introduces some challenges beyond the scope of typical image concept/object detection. The sentiment evoked from an image is subjective to each human and is dependent on many factors such as ethnic group, religion and time. An image of the twin towers would convey different sentiment before and after the 9/11 attack. The "affective gap" between the low-level visual features of an image (local or global) and its higher-level emotional content poses another challenge. Image sentiment involves a much higher level of abstraction in the human recognition process, which is difficult to model with an AI system.

3.2 State of the art

In this section we review some important works in the task of image sentiment analysis. Most the initial works, made use of hand-crafted visual features such as color and texture, before the focus shifted to convolutional neural networks (CNNs). Theoretical and empirical concepts from psychology and art theory were exploited in [Machajdik and Hanbury, 2010] to extract image features and perform emotion classification. In [Borth et al., 2013], a large scale visual sentiment ontology of Adjective Noun Pairs (ANP) was created using psychological theories and web mining. A big but noisy dataset was created by retrieving images of each ANP from Flickr. A visual concept detector library, Sentibank, was then trained to detect the presence of ANPs in images. In [Chen et al., 2014], the authors trained convolutional neural networks (CNNs) on the same dataset. The results showed the superiority of CNNs in image sentiment classification. In [Campos et al., 2015] a pre-trained CNN for the task of object classification was fine-tuned on the image sentiment analysis task. Then, a meticulous analysis of the image representations of different network layers examined their contribution to the task. In [You et al., 2015], the authors experimented with different CNN architectures and proposed a progressive training strategy, calling it PCNN, to tackle the weak labeling of the Flickr dataset, that improved the results. Their classification was on positive/negative polarity, as opposed to ANPs. Moreover, they introduced a small but human-annotated dataset consisting of 1269 images from twitter. Deep coupled adjective and noun networks were proposed in [Wang et al., 2016], to provide a mid-level representation to a third network that classifies to polarity (positive/negative), showing strong performance. In [Song et al., 2018], a visual attention mechanism is integrated to CNNs predicting the sentiment, with the results exceeding state-of-the art performance on the twitter dataset.

3.3 MIRROR approach

This section describes our first baseline approach for the task of image sentiment analysis. Deep learning and convolutional neural networks (CNNs) have been established as the state-of-the art approach and have revolutionized computer vision. Their effectiveness has drawn huge research interest and many architectures have been proposed. A typical deep convolutional neural network consists of series of convolutional layers, downsampling operators and activation functions. The weights of the layers are trained through backpropagation, with the aim of minimizing a loss function by applying an optimization algorithm. Neural networks can be trained for a specific task, but the learned weights can then be used to solve problems in similar domains, a technique also known as transfer learning. CNNs, in particular, have proved to be very effective in transfer learning applications [Oquab et al., 2014], thus our approach will be based on these principles.

For our approach, we employed the "inception" [Szegedy et al., 2015] neural network architecture, pre-trained on the Youtube-8M dataset [Abu-El-Haija et al., 2016]. The pre-trained network was used as a feature extractor for the images producing a 2048-element vector. The vector is then reduced to 1024 elements through PCA transformation. This last 1024-element representation is then used as input to the classifier, which is another neural network. This second neural network is a variant of the one-dimensional deep convolutional neural network (1D DCNN) architecture presented in [Gkalelis and Mezaris, 2020], consisting of an 1D convolutional, dropout, max-pooling and softmax layers. The same framework is used for generic visual concept detection sub-task, and is described in more detail in Section 2.1 and also in [Gkalelis and Mezaris, 2020]. Using the same feature extractor for both concept detection and sentiment analysis has a positive effect on the time performance of MIRROR's visual media analysis component, since feature extraction is performed only once for the two tasks.

The aforementioned 1D DCNN network was subsequently trained to perform classification to adjective noun pairs (ANPs). ANPs were introduced in [Borth et al., 2013] as a human-readable mid-level representation of an image. ANPs are formed from adjective and noun combinations and each one of them is associated with a certain polarity value. Polarity is a one-dimensional sentiment indicator ranging from positive to negative. We will use polarity as a binary label, being either positive or negative. We can perform the polarity classification either by training directly a binary (polarity) classifier, or, as we did in the initial experiments reported below, by classifying the images to ANPs and then mapping each ANP to its respective polarity. The mapping from ANPs to polarity was done according to [Borth et al., 2013].

3.4 Experiments, datasets, results and future work

The only dataset available with ANP labels is the Flickr dataset introduced by [Borth et al., 2013], and it is the dataset we use for our experiments. The dataset is weakly labeled as the images belonging to each ANP are the result of Flickr search queries for the respective ANP. The total size of the dataset is more than 1 million images and the total number of ANPs is 3295.

In our experiments, we follow the same workflow as in [Chen et al., 2014]. First, we discard the images belonging to ANPs with less than 120 images. This leaves us with 2125 ANPs and still over a million images. Then, the dataset is split to training and testing set. For each ANP we randomly select 20 images for testing, thus ensuring the test set is stratified. The remaining images are used for training. This step results in 1,153,411 training (96%) and 42,499 (4%) testing images.

For instantiating the 1D DCNN that is used as a classifier, described in the previous section, we used 64

Table 8: Performance comparison on ANP classification (accuracy %)

Method	all-ANPs	top-1200-ANPs
[Borth et al., 2013]	1.71	3.04
[Chen et al., 2014]	8.16	14.36
MIRROR approach	**8.30**	**14.60**

1D filters with kernel size 3 and ReLU activation for the convolutional layer. 0.7 keeprate and window of size 2 and stride 2 is applied on the dropout and max-pooling layers, respectively. The network is trained using a cross-entropy loss, Adam optimizer, exponential learning rate schedule with initial learning rate of 0.001 and decay factor 0.95 every epoch. The training is performed for 10 epochs using batch size of 512. The performance of the trained network is evaluated by measuring the accuracy of the classification. We compare against two other methods, [Borth et al., 2013] which uses hand-crafted features and SVMs for classification, and [Chen et al., 2014] which uses a deep learning technique. As far as we know there is no other published work that has reported results for the task of ANP classification. The results can be seen in Table 8. Some of the ANPs can be very abstract and/or difficult to visualize, such as ”terrible crime” or ”horrible noise”. For these ANPs, the classifier performs poorly. Thus, similar to what was done in [Chen et al., 2014], we select the top 1200 ANPs ranked by top-accuracy and calculate the accuracy score only for them.

We can see in Table 8 that our approach clearly outperforms the [Borth et al., 2013] method and also marginally outperforms [Chen et al., 2014] in both all-ANPs and top-1200. Some of the correct classification results of our method can be seen in figure 7 We also calculate the accuracy of the binary polarity classification by translating each ANP from the previous ANP classification to its respective polarity label, and the result is 63.8%. This result does not quite match the state of the art approaches such as [Wang et al., 2016]. This indicates that the indirect polarity classification method we report does not perform as well as training a binary polarity classifier on polarity-labeled images and then directly classifying the unknown-polarity images, as also preliminary results of small-scale experiments with dataset subsets have shown us.

Training direct polarity classifiers on the Flickr dataset is part of the planned future work. Also, despite being large in size, the Flickr dataset is weakly labeled. Applying techniques to de-noise the dataset can prove very important as [You et al., 2015] shows in his work. Moreover, we plan on experimenting with other datasets in the future, such as the twitter dataset, introduced in [You et al., 2015]. Finally, we plan to integrate our image sentiment analysis method to the MIRROR system in the future, via integrating sentiment in the MAS REST service discussed in Section 2.4.

Figure 7: Example results of image sentiment analysis. The detected ANP and polarity for each image can be seen next to it.

4 Automatic speech recognition

The amount and variety of multimedia- and audio-content has steadily been increasing over the past years. This concerns various Social Media channels and platforms as well as more traditional sources such as TV and radio which still play an important role in today's media and news landscape [3]. The audio contained in the documents originating from these sources contains speech as well as non-speech (background noise, music,..) and typically comes in a variety of formats, qualities, multiple languages and from a multitude of speakers. The audio conditions, speaking styles and languages, recording settings and further factors concerning the signal as well as speech pose a multitude of challenges to automatic processing, all of which need to be taken into account. In general, ASR aims to convert the speech-containing sections of such a signal into a sequence of words, representing as faithfully as possible what was said. Within MIRROR, ASR represents a key technology turning unstructured (audio) data into structured content and thus making it accessible for further processing (enrichment) and search.

4.1 Problem statement

Most of today's ASR systems are based on the so-called "Noisy-Channel-Model" originating in communication theory, whereby a sender encodes a message which is then sent via a noisy communication channel and later decoded by the recipient (The Noisy Channel Model, Figure 8). The output depends statistically on the input and the receiver's task is to decode the speech signal it receives and to retrieve the most likely words in it.

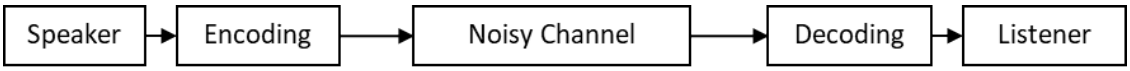

Figure 8: The noisy channel model

A speaker utters some words corresponding to a specific message. This spoken utterance is transported via a channel undergoing different types of distortion along the way which alter the original content. The ASR system is located on the recipient side of the channel. It attempts to reconstruct the original words given the distorted signal received. Statistical speech recognition systems try to reconstruct the most likely words, given the received signal, using a variety of statistical methods. All of these mechanisms try to maximize the following conditional probability:

$$P(W|A) \tag{4.6}$$

where W is the sequence of words (the written equivalent of the spoken utterance) and A is the acoustic utterance (i.e., the speech signal).

As P(W|A) cannot be determined directly, Bayes formula is applied yielding:

$$P(W|A) = (P(A|W) * P(W))/P(A) \tag{4.7}$$

Since P(A) is constant, it does not play a role in the maximization with respect to W. Finding the most likely word sequence (W*) can then be formulated as:

[3] https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2019-06/DNR_2019_FINAL_0.pdf

$$W^* = argmax_w P(W|A) = argmax_w P(A|W) * P(W) \quad (4.8)$$

From Equation 4.3 it can be inferred that two different models (knowledge-sources) can be employed for the maximization process:

- P(A|W): an acoustic model (AM), which assigns probabilities to acoustic information given particular word sequences (and their pronunciations), and
- P(W): a language model (LM) which assigns probabilities to sequences of words. The LM is built on units defined by a vocabulary, words or sub-word-units whose pronunciation(s) link the same items to the acoustic models.

The representation and processing of the speech signal forms one of the core elements of ASR systems. The AM and LM form the major knowledge sources which are employed by the ASR decoder in a search process to determine the most likely sequence of words as outlined above. These elements and their respective state-of-the-art are discussed in the following section.

4.2 State of the art

4.2.1 Speech Signal Processing

Speech signal processing deals with the spectral and time-dependent qualities of the speech signal and their connection to acoustic-phonetic phenomena in utterances produced by speakers. Due to the large variability of the speech signal, feature extraction is performed on very short durations of speech in which the audio signal is believed to have more homogeneous characteristics. Inherent signal variance (depending on microphones, channel-conditions, the physical environment where the audio is recorded) as well as speaker-dependent factors such as age, gender, emotional state, dialects, intonation or social influences might also have to be accounted for.

MFCCs, LPC or PLP-based feature representations form the basis for most of traditional automatic speech recognition systems. Various types of normalization are applied during this processing stage, typically yielding feature-vectors representing the signal across time. Typical feature vector elements include cepstral coefficients, energy-derived coefficients and delta-coefficients describing the dynamics of the signal. Often, one feature vector of 30-45 dimensions is produced for every centi-second forming the input to the acoustic modelling component of an ASR-system. Dimensionality reduction methods, such as LDA may be applied to the features before using them in the acoustic matching process [Huang et al., 2001]. More recent representations of features include i-vectors [Glembek et al., 2011], x-vectors [Snyder et al., 2018] and bottleneck features [Sainath et al., 2012].

4.2.2 Acoustic Model (AM)

Acoustic models in ASR are typically based on Hidden Markov Models (HMMs) or Neural Networks (NNs).

HMMs are based on the assumption that the speech signal possesses an inherent variation which is caused by differences in frequency as well as time-scale. HMMs are employed to describe a time-varying process, e.g. articulation of a phoneme, which is influenced by neighboring phonemes by co-articulation. In addition, the states of such models cannot be observed directly, but rather only indirectly through the sequence of

features. HMMs are generative models which can be viewed as producing feature vectors or consuming feature vectors.

HMMs owe their popularity in part to the fact that they simultaneously model the variance of the features in a temporal as well as in a spectral way, and that a sound mathematical framework to determine their parameters (for training as well as for recognition) exists. They blend in smoothly with a larger statistical framework typically found in ASR systems [Benesty et al., 2007].

Since their introduction in the late 1980s, Neural networks have also been used in many aspects of speech recognition such as phoneme classification, isolated word recognition [4], and speaker adaptation. As opposed to HMMs, neural networks make no assumptions about the statistical properties of the features. When used to estimate the probabilities of a speech feature segment, neural networks allow discriminative training in a natural and efficient manner.

Deep Neural Networks (DNNs) were proposed as an alternative technique to provide large vocabulary speech recognition, and have been most popular type of architecture successfully used as an acoustic model since 2010. Recent adaptations of NNs, Recurrent Neural Networks(RNNs) [Sak et al., 2014] and Time Delay Neural Networks (TDNNs) [Peddinti et al., 2015], have been shown to be able to identify latent temporal dependencies and therefore can perform continuous speech recognition, although the computational cost involved makes this process slower. Overall, the use of NNs for automatic speech recognition has gained increasing interest both by the academic community as well as the industry. During the recent decade, the two major conferences on signal processing and speech recognition, IEEE-ICASSP and Interspeech, have seen near exponential growth in the numbers of accepted papers on the topic of deep learning for speech recognition. More importantly, all major commercial speech recognition systems nowadays are based on deep learning methods.

4.2.3 Language Model (LM)

Language models in the area of large vocabulary ASR are mostly based on statistical models, such as N-gram models of words or word-forms. These models represent the probability of sequences of words (the component P(w) in Equation 2) and are very flexible regarding the grammaticality or semantics of a sequence of words (as a matter of fact, in unplanned speech, many sentences are either grammatically incorrect, incomplete or both). A local context is used to determine whether a word is probable or not to appear in the particular context. Typical context lengths are 2-5 words (preceding words), as words further away are not expected to exert much influence over the word under consideration. This assumption amounts to a Markov chain of the same length. The different context lengths are usually combined, yielding models of mixed complexity.

N-gram language models are used to determine the probability of a word given the local context. In a 3-gram language model, the word context in the previous sentence would thus be determined like this: P(context | the local)

A sequence of words can then be estimated by combining individual estimates:

$$P(w_1, w_2, w_3...w_4) = P(w_1) * P(w_2|w_1) * P(w_3|w_1, w_2) * ... * P(w_n|w_{n-2}, w_{n-1}) \quad (4.9)$$

The probabilities required for these estimates are determined by analyzing large text corpora (hundreds

[4] https://en.wikipedia.org/wiki/Speech_recognition#cite_note-45

of millions of words of running text) . Different methods to smooth the thus obtained probability distributions exist. Probabilities of unseen events are estimated from the text corpora and distributions smoothed accordingly. Events detected once in the corpus often serve in determining probabilities of unseen events. Smoothing methods such as Witten-Bell, Good-Turing or Kneser-Ney smoothing are commonly applied [Chen and Goodman, 1996].

The creation of language models is tightly coupled with the selection of the underlying vocabulary. Often, individual word-forms are regarded as separate entities for the language model (full-forms). However, it is also possible to use the words' lemmata as units or to group words into classes. Lemmatization requires a morphological analysis of the text corpora used to create language models. Full-forms work very well for languages with low levels of inflection (e.g. English), but encounter difficulties when used for languages with high inflection (e.g. Russian) or are completely unusable for agglutinative languages (e.g. Turkish). The unit for language modelling thus is determined by the structure of the language under consideration and the amount and types of training data available for model creation. Various methods to adapt existing LMs to a particular domain or use exist [Bacchiani and Roark, 2003].

4.2.4 Search Process

The decoder matches an incoming stream of audio features (produced by the feature extraction component(s) as described above) against a set of acoustic- and language-models and produces as its result the most likely sequence of words representing these acoustic features. This process corresponds to maximizing the numerator in Equation 4.2 above. Typically, the audio stream is first segmented to yield a section of homogeneous audio of manageable length. Ideally, each resulting segment contains speech only from one single speaker and one particular condition (typically a few seconds of audio), which can be used for adaptation and normalization operations. Likewise, these utterances form a natural unit for speaker-identification or speaker-clustering purposes [Jin et al., 1999].

During search, P(A|W) is modelled using the acoustic model while P(W) is modelled by the language model component of the decoder. Ideally, all possible sequences of words would be searched and corresponding probabilities assigned. However, due to the sheer size of the vocabulary (typically several hundreds of thousands of word-forms) and the resulting possible combinations, this is not feasible and certain approximations have to be made. In practice, the search is only carried out on a small subspace of all possible word sequences, with various ways of pruning the search-space along the way. Words and their pronunciations are aligned to the incoming audio-features in a dynamic process. The most common approach is a time-synchronous search using the Viterbi-algorithm [Rabiner and Juang, 1993]. During this process, a set of hypotheses is followed in parallel with pruning applied at several stages to keep the search-space manageable. The acoustic and language model scores are applied jointly to yield scores for the hypotheses. Stack-based decoders or decoders based on the A* algorithm form one alternative scheme of decoding.

Automatic speech recognition may be performed in a single pass or in multiple passes. Multiple passes lend themselves to a more modular approach, allowing to use more complex and more powerful models in a step-by-step manner as well as allowing to perform the complete process in a timely and parallelized manner. A further advantage of multi-pass architectures is that analysis, replacements and extensions of existing sub-systems can be undertaken in a flexible way.

4.3 MIRROR approach and relevance of ASR to MIRROR

As outlined above, the amount of multimedia and audio content has been increasing constantly over the past decades. To an already increasing amount of multimedia data coming from traditional sources, such as TV, radio or newspapers , a host of social media platforms has added tremendous amounts of further multimedia data. With the rise of social media and portable devices (especially of mobile phones), people have turned from mere consumers of information to active prosumers, producing (multimedia) content continuously and in a variety of settings. An always-on mentality caused users to consume information in near-real-time and to produce multimedia information at the same speed.

Regarding the topics addressed by MIRROR – perceptions and misperceptions of Europe and Europeans in the eyes of (non-European) migrants and citizens considering migration – multimedia content and the perceptions it creates play a fundamental role. The exact mechanisms, importance of platforms and media types, issues of credibility and more are investigated by WP7. However, it can be assumed that these kinds of contents have a strong impact on the perceptions of Europe by a large audience.

The role of ASR within MIRROR is thus to allow the project to tap into this vast ocean of information contained in multimedia and audio content. ASR allows to transcribe the (unstructured) data from TV programs and reports, press-conferences, radio-programs or YouTube videos. By converting the audio into a sequence of words (into text), further downstream technologies for the enrichment of this content can be applied (e.g. entities mentioned or the topics talked about in the audio can be detected).

The ASR engine itself is language-agnostic whereas the models are language (and domain) dependent. To account for content in different languages, models need to be trained and adapted for the particular setting. It is planned to train models for languages spoken in COO (counties of origin), COT (countries of transition) as well as COD (countries of destination) in order to be able to access content from different perspectives and presented to different audiences. Textual resources (from news media as well as from SM) will be employed in a semi-supervised manner to adapt/extend the ASR models accordingly.

Migration-related-semantic-concepts (MRSCs) play a special role in the above scheme, as they form key elements in the detection of misperceptions among groups of citizens. The representation of MRSCs in spoken language (the terminology, vocabulary, phraseology associated with them) are expected to be integrated into this process.

4.4 Experiments, datasets, results and future work

The MIRROR ASR component, based on the SAIL LABS Media Mining Indexer, uses an acoustic model that are represented as ”chain” models and language model represented as a deterministic Finite-state transducers (FST). The chain model used is a type of conventional Time Delayed Neural Network (TDNN), having a 3-fold reduced frame rate at the output of the Deep Neural Network (DNN), and optimizing a different objective function (the log-probability of the correct phone sequence instead of a frame-level objective). The reduced frame rate, which requires a modified Hidden Markov Model (HMM) topology, results in faster decoding and improved accuracy compared to those of conventional DNN-HMMs. The language model is derived from standard n-gram calculation schemes and represented as a Weighted Finite-state transducer (WFST). The acoustic model, pronunciation model, context-dependency information and the language model are combined together to form a decoding graph which is used for generating output lattices, from which N-best hypotheses can be extracted. Decoding is done in near real-time and decoding parameters can be adjusted to provide either faster or more accurate results.

Currently, ASR models are available for the following languages relevant to MIRROR: Arabic (MAS, Levantine and Egyptian), English, Farsi, German, Greek, Italian, Pashto, Turkish and Urdu. These models will be used to transcribe content in the respective language. Furthermore, all of these models are employed in the Media Mining System of SAIL LABS which serves as a data-provider (in a DaaS manner) to the MIRROR project. The above ASR models have been built with the aim to use them in transcribing general broadcast-news contents (including topics of politics, the economy, sports and gossip). However, as the domain of MIRROR is only partly covered by these topics, the ASR models need to be adjusted/adapted to the migration domain and in particular to domains connected with the different kinds of MRSCs (these need to be detected in audio-visual content and thus need to be present in the ASR models).

Work on ASR within project year 1 has consequently focused on adaptation of ASR models for the MIRROR domain and on the creation of an environment which allows to perform this adaptation in a semi-automatic manner. Adjusting and extending the vocabulary of relevant languages has been achieved by including terminology and sources relevant to the domain of migration. This line of work concerns the addition of media-sources relevant to migration and the subsequent automatic extraction of relevant terminology from these sources (described in [Dikici et al., 2019]) as well as a connection to the Named Entity Recognition (NER) component within the SAIL LABS system, where newly added NE are also included into the ASR vocabulary and LM.

Figure 9 Migration relevant terminology below provides some examples for English. Several entries have been in the vocabulary before, but specific terminology such as “repatriation” or “influx” have been added to the model.

Figure 9: Migration relevant terminology

Figure 10 Repatriation recognized in TV program on RT below shows a sample where repatriation has successfully been recognized on TV content from RT.

The ASR models for English, German, French, Italian and Spanish have been re-built in this manner during year 1 (most recently on a weekly basis). Further improvements to the underlying base technology have

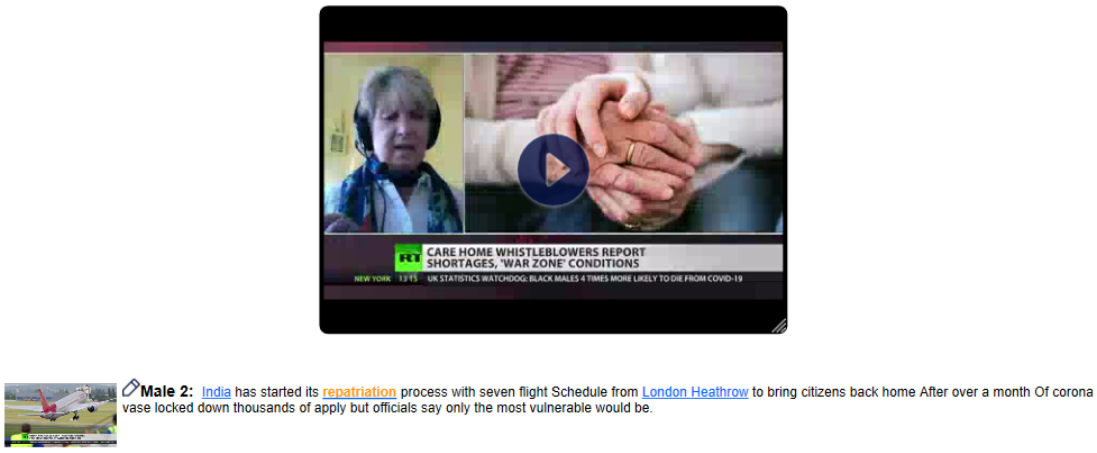

Figure 10: Repatriation recognized in TV program on RT

been made by adapting the wsj and librispeech recipes of version v20200309 of KALDI [5] [Povey et al., 2011].

The ASR component itself provides a C/C++/C#-API and has been packaged into a docker-component, exposing a REST-API. Currently, the ASR models are packaged together with the ASR-engine, resulting in one docker-component per language. In the future, we plan to make externalize models on a shared -object store and only load them on demand. Future work on the ASR itself will be dedicated to the semi-supervised improvement of ASR, further continuous integration of advances from KALDI and the integration of the MRSCs for all languages relevant to the project.

[5] https://kaldi-asr.org/doc/about.html

5 Conclusions

In this deliverable we presented the first set of the multimedia analysis methods developed in MIRROR as part of WP5. For multimedia annotation, three distinct approaches were presented. The diverse visual annotations that these approaches generate will enable multiple forms of multimedia retrieval in the MIRROR platform. For MRSC detection, the lack of labeled datasets has led us to adopt a zero-shot learning paradigm. The initial evaluation showed that our method achieves satisfactory performance but there is also room for improvement. For image captioning, we plan on continuing experimenting to improve the performance of our baseline method. For image sentiment analysis we presented our first approach that shows promising results for adjective noun pair classification. Future work will focus on improving performance, especially on polarity classification. For automatic speech recognition, the initial MIRROR approach promises to transcribe audio sources with state of the art accuracy and speed. Several languages are already supported, and more models will be trained for additional languages of interest to MIRROR. Future work will include semi-supervised improvement of ASR and the integration of MRSCs. On the integration side, a REST service (MAS) has been implemented and set-up for MIRROR and already hosts a subset of the WP5 visual analysis tools. In the future, it will expand to also include all the WP5 visual sentiment analysis methods. A separate service is also available for ASR: the ASR component has been packaged into a docker-component and can be accessed through its REST API.

6 References

[99firms, 2019] 99firms (2019). Facebook Video Statistics. https://99firms.com/blog/facebook-video-statistics/#gref. Last visited 12-5-2020.

[Abu-El-Haija et al., 2016] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *CoRR*, abs/1609.08675.

[Akata et al., 2016] Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. (2016). Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438.

[Apostolidis et al., 2018] Apostolidis, K., Apostolidis, E., and Mezaris, V. (2018). A motion-driven approach for fine-grained temporal segmentation of user-generated videos. In Schoeffmann, K., Chalidabhongse, T. H., Ngo, C. W., Aramvith, S., O'Connor, N. E., Ho, Y.-S., Gabbouj, M., and Elgammal, A., editors, *MultiMedia Modeling*, pages 29–41, Cham. Springer International Publishing.

[Arestis-Chartampilas et al., 2015] Arestis-Chartampilas, S., Gkalelis, N., and Mezaris, V. (2015). GPU accelerated generalised subclass discriminant analysis for event and concept detection in video. In *Proc. ACM MM*, pages 1219–1222, Brisbane, Australia.

[Aslam, 2020] Aslam, S. (2020). YouTube by the Numbers: Stats, Demographics & Fun Facts. https://www.omnicoreagency.com/youtube-statistics/. Last visited 12-5-2020.

[Awad et al., 2019] Awad, G., Butt, A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Delgado, A., Zhang, J., Godard, E., Diduch, L., Smeaton, A. F., Graham, Y., Kraaij, W., and Quénot, G. (2019). Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID 2019*. NIST, USA.

[Bacchiani and Roark, 2003] Bacchiani, M. and Roark, B. (2003). Unsupervised language model adaptation. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 1, pages I–I.

[Benesty et al., 2007] Benesty, J., Sondhi, M. M., and Huang, Y. A. (2007). *Springer Handbook of Speech Processing*. Springer-Verlag, Berlin, Heidelberg.

[Bengio, 2009] Bengio, Y. (2009). Learning deep architectures for ai. *Foundations*, 2:1–55.

[Borth et al., 2013] Borth, D., Ji, R., Chen, T., Breuel, T., and Chang, S.-F. (2013). Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, page 223–232, New York, NY, USA. Association for Computing Machinery.

[Campos et al., 2015] Campos, V., Salvador, A., Jou, B., and Giró-i Nieto, X. (2015). Diving deep into sentiment: Understanding fine-tuned cnns for visual sentiment prediction.

[Carreira and Zisserman, 2017] Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. In *Proc. IEEE CVPR*, pages 4724–4733, Honolulu, HI, USA.

[Chen and Dolan, 2011] Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

[Chen et al., 2017] Chen, S., Chen, J., Jin, Q., and Hauptmann, A. (2017). Video captioning with guidance of multimodal latent topics. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1838–1846.

[Chen and Goodman, 1996] Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, page 310–318, USA. Association for Computational Linguistics.

[Chen et al., 2014] Chen, T., Borth, D., Darrell, T., and Chang, S. (2014). Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks.

[Deng and Yu, 2014] Deng, L. and Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dikici et al., 2019] Dikici, E., G., B., and J., R. (2019). *The SAIL LABS Media Mining Indexer and the CAVA Framework*. Proc. Interspeech 2019, 4630-4631.

[Donahue et al., 2017] Donahue, R., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., and Darrell, T. (2017). Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):677–691.

[Dong et al., 2018] Dong, J., Li, X., and Snoek, C. G. M. (2018). Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia (TMM)*, 20(12):3377–3388.

[Dong et al., 2019] Dong, J., Li, X., Xu, C., Ji, S., He, Y., Yang, G., and Wang, X. (2019). Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9346–9355.

[Faghri et al., 2018] Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. (2018). Vse++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Feichtenhofer et al., 2016] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *IEEE CVPR*, pages 1933–1941.

[Galanopoulos and Mezaris, 2020] Galanopoulos, D. and Mezaris, V. (2020). Attention mechanisms, signal encodings and fusion strategies for improved ad-hoc videosearch with dual encoding networks. In *Proceedings of the 2020 ACM on International Conference on Multimedia Retrieval (ICMR'20)*, ICMR '20. ACM.

[Gallo et al., 2019] Gallo, F., Russo, G., Elejalde, E., Shaltev, M., Backfried, G., Dikici, E., Mezaris, V., Pournaras, A., Pia, J., and Bonnici, M. (2019). MIRROR Architecture and Integration Plan.

[Gan et al., 2015] Gan, C., Wang, N., Yang, Y., Yeung, D., and Hauptmann, A. G. (2015). Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proc. IEEE CVPR*, pages 2568–2577, Boston, MA, USA.

[Girdhar et al., 2017] Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., and Russell, B. C. (2017). Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proc. IEEE CVPR*, pages 3165–3174, Honolulu, HI, USA.

[Gkalelis and Mezaris, 2020] Gkalelis, N. and Mezaris, V. (2020). Subclass deep neural networks: Re-enabling neglected classes in deep network training for multimedia classification. In *Proc. Int. Conf. MultiMedia Modeling*, volume 11961, pages 227–238, Daejeon, South Korea.

[Glembek et al., 2011] Glembek, O., Burget, L., Matejka, P., Karafiát, M., and Kenny, P. (2011). Simplification and optimization of i-vector extraction. pages 4516 – 4519.

[Gygli, 2017] Gygli, M. (2017). Ridiculously fast shot boundary detection with fully convolutional neural networks. *CoRR*, abs/1705.08214.

[Habibian et al., 2017] Habibian, A., Mensink, T., and Snoek, C. G. (2017). Video2vec embeddings recognize events when examples are scarce. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(10):2089–2103.

[Hara et al., 2017] Hara, K., Kataoka, H., and Satoh, Y. (2017). Learning spatio-temporal features with 3d residual networks for action recognition. In *Proc. IEEE ICCV Workshops*, pages 3154–3160, Venice, Italy.

[Hara et al., 2018] Hara, K., Kataoka, H., and Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proc. IEEE CVPR*, pages 6546–6555, Salt Lake City, UT, USA.

[Hershey et al., 2017] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., and Wilson, K. W. (2017). CNN architectures for large-scale audio classification. In *Proc. IEEE ICASSP*, pages 131–135, New Orleans, LA, USA.

[Huang et al., 2001] Huang, X., Acero, A., Hon, H.-W., and Reddy, R. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, USA, 1st edition.

[Hussein et al., 2019] Hussein, N., Gavves, E., and Smeulders, A. W. M. (2019). Timeception for complex action recognition. In *Proc. IEEE CVPR*, pages 254–263, Long Beach, CA, USA.

[Hussein et al., 2020] Hussein, N., Gavves, E., and Smeulders, A. W. M. (2020). PIC: permutation invariant convolution for recognizing long-range activities. *arXiv:2003.08275*.

[Ji et al., 2013] Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231.

[Jin et al., 1999] Jin, H., Kubala, F., and Schwartz, R. (1999). Automatic speaker clustering. proc. of the 1999 darpa speech recognition workshop.

[Jin et al., 2016] Jin, Q., Chen, J., Chen, S., Xiong, Y., and Hauptmann, A. (2016). Describing videos using multi-modal fusion. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1087–1091.

[Karpathy et al., 2014] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Li, F. (2014). Large-scale video classification with convolutional neural networks. In *Proc. IEEE CVPR*, pages 1725–1732, Columbus, OH, USA.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA. Curran Associates Inc.

[Lampert et al., 2014] Lampert, C. H., Nickisch, H., and Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465.

[Lee et al., 2018] Lee, J., (Paul) Natsev, A., Reade, W., Sukthankar, R., and Toderici, G. (2018). The 2nd YouTube-8M large-scale video understanding challenge. In *ECCV Workshops*, Munich, Germany.

[Li et al., 2016] Li, Y., Song, Y., Cao, L., Tetreault, J., Goldberg, L., Jaimes, A., and Luo, J. (2016). Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650.

[Ma et al., 2018] Ma, Z., Chang, X., Xu, Z., Sebe, N., and Hauptmann, A. G. (2018). Joint attributes and event analysis for multimedia event detection. *IEEE Trans. Neural Networks Learn. Syst.*, 29(7):2921–2930.

[Machajdik and Hanbury, 2010] Machajdik, J. and Hanbury, A. (2010). Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 83–92, New York, NY, USA. Association for Computing Machinery.

[Markatopoulou et al., 2017] Markatopoulou, F., Galanopoulos, D., Mezaris, V., and Patras, I. (2017). Query and keyframe representations for ad-hoc video search. In *Proceedings of the 2017 ACM International Conference on Multimedia Retrieval*, ICMR '17, pages 407–411. ACM.

[Markatopoulou et al., 2015] Markatopoulou, F., Ioannidou, A., and Tzelepis, C. e. a. (2015). Iti-certh participation to trecvid 2015. In *Proceedings of the TRECVID 2015 Workshop, Gaithersburg, MD, USA, Nov. 2015*.

[Markatopoulou et al., 2016a] Markatopoulou, F., Mezaris, V., and Patras, I. (2016a). Deep multi-task learning with label correlation constraint for video concept detection. In *Proc. ACM MM*, pages 501–505, Amsterdam, The Netherlands.

[Markatopoulou et al., 2016b] Markatopoulou, F., Mezaris, V., and Patras, I. (2016b). Online multi-task learning for semantic concept detection in video. In *Proc. IEEE ICIP*, pages 186–190, Phoenix, AZ, USA.

[Markatopoulou et al., 2019] Markatopoulou, F., Mezaris, V., and Patras, I. (2019). Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. *IEEE Trans. Circuits Syst. Video Techn.*, 29(6):1631–1644.

[Markatopoulou et al., 2013] Markatopoulou, F., Moumtzidou, A., Tzelepis, C., and et al. (2013). Iti-certh participation to trecvid 2013. In *Proceedings of the TRECVID 2013 Workshop, Gaithersburg, MD, USA, Nov. 2015*.

[Mettes et al., 2016] Mettes, P., Koelma, D. C., and Snoek, C. G. M. (2016). The imagenet shuffle: Reorganized pre-training for video event detection. In *Proc. ACM ICMR*, pages 175–182, New York, New York, USA.

[Miech et al., 2017] Miech, A., Laptev, I., and Sivic, J. (2017). Learnable pooling with context gating for video classification. *arXiv:1706.06905*.

[Mikolov et al., 2013] Mikolov, T., Corrado, G., Chen, K., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, Workshop Track Proceedings*, ICLR '13.

[Ng et al., 2015] Ng, J. Y., Hausknecht, M. J., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proc. IEEE CVPR*, pages 4694–4702, Boston, MA, USA.

[Norouzi et al., 2013] Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., and Dean, J. (2013). Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.

[Oquab et al., 2014] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Peddinti et al., 2015] Peddinti, V., Chen, G., Manohar, V., Ko, T., Povey, D., and Khudanpur, S. (2015). Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 539–546.

[Pittaras et al., 2017] Pittaras, N., Markatopoulou, F., Mezaris, V., and Patras, I. (2017). Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In *Proc. Int. Conf. MultiMedia Modeling*, volume 10132, pages 102–114, Reykjavik, Iceland.

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

[Qiu et al., 2017] Qiu, Z., Yao, T., and Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *Proc. IEEE ICCV*, pages 5534–5542, Venice, Italy.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., USA.

[Rohrbach et al., 2015] Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B. (2015). A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *Int. Journal of Comput. Vision*, 115(3):211–252.

[Sainath et al., 2012] Sainath, T. N., Kingsbury, B., and Ramabhadran, B. (2012). Auto-encoder bottleneck features using deep belief networks. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4153–4156.

[Sak et al., 2014] Sak, H., Senior, A. W., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proc. NIPS*, pages 568–576, Montreal, Quebec, Canada.

[Snyder et al., 2018] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

[Song et al., 2018] Song, K., Yao, T., Ling, Q., and Mei, T. (2018). Boosting image sentiment analysis with visual attention. *Neurocomputing*, 312.

[Sun et al., 2019] Sun, L., Li, B., Yuan, C., Zha, Z., and Hu, W. (2019). Multimodal semantic attention network for video captioning. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1300–1305. IEEE.

[Szegedy et al., 2015] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

[Tran et al., 2015] Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE ICCV*, pages 4489–4497, Santiago, Chile.

[Varol et al., 2018] Varol, G., Laptev, I., and Schmid, C. (2018). Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1510–1517.

[Venugopalan et al., 2015] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.

[Wang et al., 2017] Wang, B., Yang, Y., Xu, X., Hanjalic, A., and Shen, H. T. (2017). Adversarial cross-modal retrieval. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 154–162.

[Wang and Schmid, 2013] Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *Proc. IEEE ICCV*, pages 3551–3558, Sydney, Australia.

[Wang et al., 2016] Wang, J., Fu, J., Xu, Y., and Mei, T. (2016). Beyond object recognition: Visual sentiment analysis with deep coupled adjective and noun neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 3484–3490. AAAI Press.

[Wang et al., 2018a] Wang, X., Wang, Y.-F., and Wang, W. Y. (2018a). Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. *arXiv preprint arXiv:1804.05448*.

[Wang et al., 2018b] Wang, X., Yu, F., Wang, R., Ma, Y., Mirhoseini, A., Darrell, T., and Gonzalez, J. E. (2018b). Deep mixture of experts via shallow embedding. *arXiv:1806.01531*.

[Xian et al., 2017] Xian, Y., Schiele, B., and Akata, Z. (2017). Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591.

[Xu et al., 2016] Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

[Yao et al., 2019] Yao, T., Pan, Y., Li, Y., and Mei, T. (2019). Hierarchy parsing for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2629.

[You et al., 2015] You, Q., Luo, J., Jin, H., and Yang, J. (2015). Robust image sentiment analysis using progressively trained and domain transferred deep networks.

[Zhang and Peng, 2019] Zhang, J. and Peng, Y. (2019). Object-aware aggregation with bidirectional temporal graph for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8327–8336.

[Zhen et al., 2019] Zhen, L., Hu, P., Wang, X., and Peng, D. (2019). Deep supervised cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10394–10403.