2018

USING MACHINE LEARNING TO CREATE PERSONALISED SNACKABLE CONTENT

Evangelos Stromatias, Karthik Yadati, Martin Prins & Joost de Wit

Media Distillery, The Netherlands

ABSTRACT

Despite having a large amount of media content, the user experience of pay TV services is often not on par with Over-The-Top (OTT) Video-on-Demand (VoD) offerings, which can cater to current audiences with short snippets and binge watch capabilities. To address this, some TV providers offer short-form content by manually “cutting” the linear content into VoD assets. However, this is often neither feasible nor a scalable solution. Moreover, research shows consumers are struggling to discover new content leading to frustration. Machine learning (ML) algorithms and Deep Learning (DL) in particular have gained tremendous popularity over the past years because they have performed comparable to, and in some cases superior to, human experts in object and speech recognition tasks. In this paper, we present our platform which utilises state-of-the-art ML algorithms for the real-time analysis of thousands of hours of multilingual multimedia content, including television and VoD. These analyses enable us to obtain rich metadata from the content of videos and suggest small chunks of personalised content (”snacks”) to users based on their preferences.

INTRODUCTION

Over the past decade, consumer viewing habits have changed drastically. TV operators and broadcasters, while they produce a lot of original content, are losing ground to OTT VoD service providers, that provide short snippets and binge-watching capabilities among others. Moreover, data shows that the younger generations spend half their time-consuming VoD content, an increase of more than 100% from 2010 until 2017 [1]. To compete with the OTT VoD offerings, TV service providers attempt to manually cut linear content into smaller snippets and recommend these to viewers (like NPO Start[1]). This, however, is a costly and time-consuming procedure [2].

Interestingly enough, despite their differences both TV providers and OTT VoD service providers face the same problem: viewers find content discovery still very challenging [1, 3]. Recent reports show that viewers spend on average 1 hour per day searching for content and this number is expected to increase as more content becomes available [1]. Approximately 70% of viewers will prefer on-demand and catch-up services over linear TV content [1]. Meanwhile, research shows that providing searching capabilities and recommendations to improve the viewer's engagement [3]. These findings suggest that both content segmentation and recommendation must happen in an efficient and automated way to keep up with a large amount of content available and satisfy the customer need.

[1] http://www.npo.nl

Deep Learning (DL), a subfield of machine learning (ML) in Artificial Intelligence (AI), has been successfully applied to solve cognitive tasks that were previously thought to be solvable only by human experts, thus gaining tremendous popularity over the past decade [4]. We have developed a multimedia content analysis platform which leverages ML algorithms to analyse thousands of hours of video and audio content in real-time. Some of the algorithms we utilize enable us to convert speech to text, recognize faces, identify objects, detect text and logos. The result of these algorithms enables us to understand and search in video and audio content and create rich metadata.

In this paper, we describe our developments in rich metadata extraction and investigate new media applications based on this detailed understanding of videos, such as in-content search, contentbased recommendations and snackable content. We also present three use cases for our platform.

RELATED WORK

Tech giants like Google[2], Amazon[3], IBM[4] and Microsoft[5] have their own cloud-based platforms and offer ready-to-use ML solutions such as face recognition, speech recognition and machine translation to name a few. However, these off-the-shelf ML solutions suffer from several shortcomings that render them unsuitable for particular tasks, including content understanding. Some of these shortcomings include (a) communication overhead; (b) expensive to use; (c) unable to deal with peculiarities in certain production data (e.g., interlacing) and (d) difficult to customize to specific customer needs. For these reasons, a lot of companies develop their own in-house solutions for personalised content recommendation systems.

YouTube introduced their automatic speech recognition (ASR) system in 2009 for automatic video captioning. The automatically generated subtitles enable YouTube to search within videos and recommend specific segments, referred to as “snippets”, to their users. YouTube’s current personalised recommendation engine comprises a series of ML algorithms and data shows that 70% of the time users spend watching videos on YouTube is due to their personalised recommendation engine [5].

Spotify uses various ML algorithms for recommending music content to their users. These algorithms range from collaborative filtering using 1.5 billion user-generated playlists to applying natural language processing (NLP) [6] on blogs and news about artists. Also, Spotify has experimented with DL models for content-based music recommendation, using not the provided song metadata but the raw audio signal of the song itself [6, 7].

Netflix’s content recommendation pipeline consists of multiple ML algorithms [8] and they also employ ML for presenting the suggested content in a personalised manner to their customers [9]. They reported that 80% of their content consumption comes from their recommender system and the rest 20% of their search service. They estimated that the yearly savings due to the combined effect of personalisation and recommendations are above $1B per year [8]. These findings demonstrate the importance of personalised content recommendation.

[2] https://cloud.google.com/products/machine-learning/

[3] https://aws.amazon.com/machine-learning/

[4] https://www.ibm.com/cloud/machine-learning

[5] https://azure.microsoft.com/en-us/services/cognitive-services/

PERSONALISED SNACKABLE CONTENT

Our goal is to create and offer bite-sized content, in the form of short snippets extracted from longform audio-visual content (which we call snackable content), to cater to those consumers that either have a couple of minutes to spare, already know what specific content they would want to view or those who want to spend the least amount of time looking for relevant content. We aim to leverage the vast collections of long-form audio-visual broadcast content that broadcasters and TV operators have available.

There are multiple ways in which one can create snacks from long-form media content. Snacks might have a different meaning depending on the use-case and the type of video. For example, a person consuming a news broadcast might think of snacks as the different news stories being discussed. Another example could be an organization monitoring how many times their brand is being mentioned in a radio broadcast. For them, snacks could be all the snippets mentioning their brand in the long radio broadcast. An important aspect of creating snackable content is to define the criteria on which the snacks need to be created, which requires the extraction of rich metadata.

Extracting Rich Metadata From Broadcasted Content

In this section, we take a look at the technologies that can now be used to create rich metadata from audio-visual content which we will exploit to create snackable content. However, due to size limitation, we will only describe a number of them and provide a comparison with the state-of-theart. We focus on the technologies that until recently were too complex or computationally intensive to use in an operational setting. We also share some of our initial findings in our research.

Face Recognition

Face recognition algorithms are capable of identifying a person by his face given a digital image or a video frame. State-of-the-art models in academic research are mainly focused on face verification and identification and report results on the labelled faces in the wild (LFW) [10] test set. Face verification aims at determining whether two given faces belong to the same person, whereas face identification compares a query face against a database of faces with the aim of identifying the queried person. For our use case, however, the model should be able to simultaneously detect if a person is unknown (does not belong in a database of faces we would like to detect) and also provide a label for each detected face.

Figure 1 shows our face recognition pipeline, which consists of the following subsystems: face detection, facial landmark extraction and face alignment, a deep convolutional neural network that generates a vector describing the query face and finally our proprietary face classifier. The most important component of the pipeline is the deep neural network. For face verification and identification, the face embeddings are used to detect if two faces belong to the same person by comparing the distance between them using a distance metric such as Euclidean distance or cosine similarity.

Our deep neural network architecture is based on [11], but we did not use the triplet-loss method as in the original paper. Instead, we used the centre loss function [12] which aims at simultaneously learn a centre for each class and penalize the distances between the deep features and their corresponding class centres. We trained our deep learning model on MS-Celeb-1M dataset [13], the largest publicly available labelled faces dataset consisting of 10 million images of 100 thousand celebrities. During training, various augmentations were employed such as random rotations and image flipping.

On the face verification task, our deep neural network achieves an accuracy of 99.2% on LFW, while the current state-of-the-art [11] achieves a score of 99.63% and is trained on 260 million faces. A summary of the results can be seen in Table 1. While there is still a gap in accuracy between our model and the current state-of-the-art we have found that models with high scores in academic datasets do not always translate to better performance in broadcasted data.

Since we are interested in face classification and not verification or identification, we have developed our own proprietary classifier that is trained on the face embeddings of individuals that need to be detected in our platform. By detecting various individuals, our users can search for specific people, and we provide them with a snippet where that particular person appears in the video.

In addition to the challenges faced by face recognition in images (pose, lighting, occlusion etc.), we are also faced with challenges like motion blur and interlacing because of the nature of our data (broadcast video). We aim to overcome these issues with more data augmentations during training in the future. Moreover, a face recognition system trained on a set of people can only recognize these people in unseen videos. There is still limited research in open set face recognition, where you first recognize whether a face is known or unknown and then proceed towards recognizing the known faces. Finally, since we have a lot of unlabelled video data, we are investigating methods such as unsupervised face clustering.

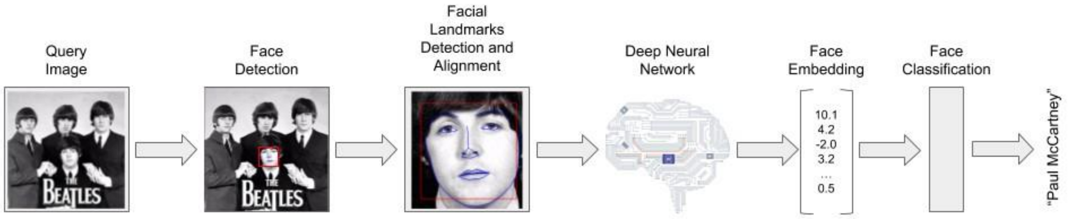

Figure 1 – Our face recognition pipeline.

Table 1 – Comparison of our deep neural network on face verification task on the LFW dataset with the state-of-the-art. The table includes number of training examples, number of models since some implementations utilise an ensemble architecture, whether an alignment method was used and distance metric used.

Model	# Train data	# Models	Align Train/Test	Face Verification	LFW (%)
FaceNet [11]	260M	1	No/No	L^2 distance	**99.63**
DeepFace [28]	4M	4	3D	Dot prod., X2, Log. reg.	98.37
Parkhi et al. [29]	2.6M	1	No/2D	L^2 distance	99.13
our deep neural network	10M	1	2D/2D	L^2 distance	99.20

Text in the wild

Text in the wild tackles the problem of text spotting in visual content: localising and recognising text in images. The input is an image that could be a scene (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image in a video (for example from a television broadcast). This information can be very useful in identifying events in a video.

Our text spotting pipeline consists of a text detection method followed by a text recognition method and can be observed in Figure 2. The proposed model is inspired by Faster-RCNN [14] and utilizes the VGG-16 architecture trained on ImageNet [15], which is a popular CNN designed for object classification in natural scenes. We remove all the fully connected layers since we are interested in using this architecture only for feature extraction. The convolutional features of VGG-16 are then fed to a text proposal network which comprises 2 fully connected layers that slide through the extracted feature maps and output a set of textness scores and regression scores for each predefined anchor box on every sliding window location of the feature maps.

The text proposals are sorted based on their textness scores, and the top n are passed to the text detection network. The text detection network is a multi-layer perceptron (MLP) that aims to classify whether a proposed region is text or not. Afterwards, all word proposals are cropped from the original image and are passed to the text recognition network, which predicts the most likely sequence of characters depicted in the word image. Our text recognition is an encoder-decoder model with Gated Recurrent Units (GRU) [16] with a soft-attention mechanism [17].

Because most available datasets for text spotting are too small and deep learning models require a lot of data to prevent overfitting, two synthetic datasets were created recently. Synth90k [18] dataset contains 9 million synthetically generated word images, of which 7.2 million are used for training, 900k for validating and 900k for testing. Synth800k [18] contains 800k training examples.

For training our text detection method, we used the Synth800k dataset and fine-tuned on the Street View Text Dataset (SVT) [19] and ICDAR03 [20] training sets. For training the text recognition method, we combine the Synth90k and Synth800k resulting in 12 million images in total. The model is trained for 2 epochs.

For evaluating our text detection method, we compute the total word recall on SVT and ICDAR. For evaluating our text recognition method, we use the cropped word images from ICDAR03 [20], ICDAR11[20], IIIT5K [21] and SVT [19].

Results for the text detection method are summarised in Table 1. Our model has a recall comparable to the state-of-the-art with far fewer proposals. On the SVT dataset, we achieve a recall of 93% with 400 proposals, while [18] achieves a 97% recall with more than 10k proposals.

Table 2 summarizes the results of our text recognition method and compares with the state-of-theart. Our model is very close to the state-of-the-art [18] while it is trained for only 1 less. We are currently in the process training for more epochs and fine-tuning our text spotting model to broadcasted TV data.

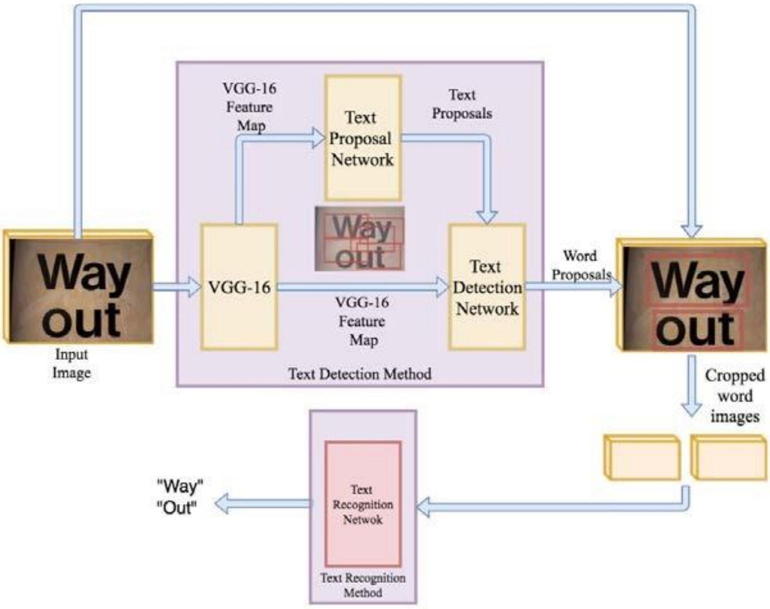

Figure 2 – High-level overview of our text spotting pipeline.

Table 2 – Evaluation of the relation between the number of word box proposals and total word recall on the Street View Text Dataset (SVT) [19] compared to state-of-the-art proposal methods.

Method	**# Proposals**	**Recall**
TextProposals [22]	17358	0.94
Jaderberg et al.[18] without (RF+CNN-reg)	>10^4	**0.97**
Jaderberg et al.[18] without CNN-reg	900	0.9480
Karaoglou et al. [23] ***MSER+TSAL, I+I, O_2+S,H***	968	0.9614
Our Text Detection Method	100	0.89
	400	0.93

Table 3 – Evaluation of the text recognition accuracy on several benchmark datasets.

Method	**Cell Type**	**Datasets**	**Epochs**	**SVT**	**IIIT$_5$K**	**ICDAR$_{03}$**	**ICDAR$_{11}$**
Lee and Osindero [24]	LSTM	Synth90k	-	80.7%	78.4%	88.7%	**90.0%**
Cheng et al. [25] baseline	LSTM	Synth90k+ Synth800k	3	**82.2%**	**83.7%**	**91.5%**	89.4%
Our Text Recognition method	GRU	Synth90k+ Synth800k	2	80.37%	83.24%	89.90%	87.51%

Logo recognition

Logo recognition involves detecting and recognizing logos of brands when they appear in the visual stream. Our logo recognizer has the capability of simultaneously detecting and recognizing a logo. This is especially valuable for brands to monitor how many times their logo appears in videos as this could be useful to measure the reach of their brand and also improve their spending on advertising. Our logo recognition model is based on the Faster-RCNN [14] architecture, which is trained on the ImageNet [15] dataset. We further fine-tuned our model using our proprietary logo dataset, which consists of six brands and a background class (negative examples). We achieve a mean precision of 98% and mean recall of 86%. For our use cases we focus on reducing the number of falsepositives. Thus we aim for higher precision than recall. Comparing with the state-of-the-art is not possible because our dataset is not publicly available for legal purposes.

Creating Snacks

In the previous section, we looked at numerous technologies that we can utilise for creating snackable content. All these technologies enable us to extract rich metadata from audio-visual signals. Now, we need to create snippets from a long video, and these should be coherent. For example, we cannot cut the video at a point where a speaker in the video is in the middle of a sentence. To achieve this, we use video shot transition detection and scene transition detection[6]. Shots are sequences of consecutive frames captured by a single camera without interruption.

[6] http://mklab.iti.gr/project/video-shot-segm

Scenes are higher level temporal snippets that correspond to the story-telling parts of the video. They are formed by grouping the detected shots into semantically coherent temporal video snippets. We have shot/scene boundaries and the metadata extracted from the video. Suppose the user wants to see all snippets where a celebrity appears in the visual stream. From the output of face recognition, we know where that particular celebrity appears. We can create a snack by cutting at the nearest shot/scene boundaries where the celebrity appears. This is a challenging task as what constitutes a scene is highly subjective, though a shot is well defined. There are also limitations on using only the visual stream, as there might be a case where things have changed visually but the speaker is still talking about the same topic in the audio channel. In future work, we will explore combining audiobased segmentation with visual segmentation.

Content Recommendations

To personalise the audio-visual content offered to users, recommendation engines are often used. Many TV service providers use (off-the-shelf) content-based recommenders that rely on metadata as their only source of information about the content. In many cases, this metadata is insufficient to get accurate recommendations because it is missing altogether or does not describe the content in enough detail, forcing some operators to have editorial teams for recommendations. By using our content understanding technologies, extensive and consistent metadata can be created automatically and on a snack-level too.

One approach to recommend content snacks is to use the extracted metadata in addition to the regular program-specific metadata such as title, host and synopsis. Detected topics, faces, logos and objects allow the recommendation engine to suggest snacks based on people, interests, brands and events by learning the user’s preference for these labels.

PAST, PRESENT, AND FUTURE USE CASES

The technology underlying Snackable Content can be used in numerous ways to create media applications. When it becomes possible to automatically create snippets and understand their content, lots of new use cases become feasible since they no longer depend on manual labour. Here, we list three use cases.

Filmstrip

The Filmstrip [26], takes the visual properties of video into account when presenting the suggested content to the user. Most video services only show a single thumbnail per asset, which is often a curated stock image identical for different episodes. This single thumbnail hardly gives a clue about the contents of the video the user is about to watch. The Filmstrip aims to give a visual summary of the entire video by selecting descriptive keyframes for each shot. These keyframes are cropped based on the length of the shot and then stitched together to a very wide image. Users can interact with the Filmstrip by swiping it back and forth, thus allowing them to seek in the video.

An initial version of the Filmstrip was developed and evaluated together with TNO[7] and Dutch news broadcaster NOS[8]. The concept was applied on NOS news bulletins that were published on their website. Users that tested it were asked to take part in a survey, which was completed by 91 respondents. 64% of the users indicated that they enjoyed the concept and wanted to continue using it to watch NOS Journaal content. 60% thought it would be valuable for other content as well [27].

[7] http://www.tno.nl

[8] http://www.nos.nl

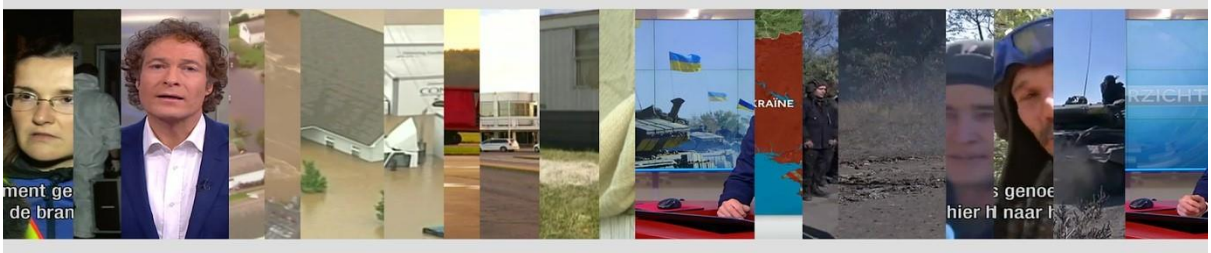

Figure 3 - Filmstrip viewport [26].

Personalised Playlist

Spotify's Daily Mix and YouTube's Autoplay are doing a great job when it comes to keeping users hooked to their service [12,11]. They create personalised playlists of songs and videos respectively, offering a lean back experience to the user. Almost no interaction is required to keep consuming content. Snackable Content allows TV service providers and broadcasters to create the same experience with the content they serve.

So far, we have contributed to two proofs-of-concept that build on this use case: Smart Radio and NewsGenius [27]. Smart Radio is currently being developed together with a news radio station in The Netherlands. The goal is to automatically create personalised podcasts and news streams that cover the topics of interest to the user. The content is created from regular news broadcasts (live) and podcasts and primarily contains only audio. Smart Radio aims to turn a linear listening experience into a personalised on-demand one.

NewsGenius aims for similar user experience but is video based. It automatically creates a news bulletin of a predefined duration (ranging from 5 to 30 minutes), covering only topics of interest to the user. The topics are determined using automatic topic detection that is applied to the speech-totext results. The user's preferences concerning these topics are learnt from the content consumption and are adapted continuously. To offer a smooth viewing experience, the order of the topics and bumpers between topic changes are carefully chosen.

Content Discovery

TV service providers often have a lot of valuable content available in their catch-up, replay and VoD catalogues, but finding what you are looking for is a big challenge. A recent report published by Ericsson shows that people spend 51 minutes searching for relevant content each day [2]. Extensive metadata is required to fulfil users' expectations, especially with the growing popularity of "voice search". A voice commands such as "show me a video of Max Verstappen overtaking Lewis Hamilton" requires deep content understanding to return relevant results. When you currently search for "Max Verstappen" in most of the TV provider's platforms, you will get "0 results found".

Another popular way to discover content is through carousels, (horizontal) lists of suggested content that is somehow related. Browsing through the items is easy and convenient using the remote control or by swiping. Common examples of carousels are "Now popular", "Recently added" and "Action Thrillers" in Netflix. With a better understanding of the content, it becomes feasible to create much more specific and potentially personalised carousels covering topics, people and other interests.

CONCLUSIONS AND FUTURE WORK

The services and underlying technologies described in this paper are still in an early phase of development. We plan to extend and improve our services in numerous ways.

First, we want to scale up the training of our algorithms by applying semi-supervised and unsupervised learning. By using semi-supervised training, humans only have to check the algorithm's suggestions, while unsupervised learning can be achieved by combining multiple modalities such as face and text recognition.

Furthermore, we plan to improve the automatic content segmentation and train it on audio-only content too. Currently, features such as shot boundaries are used, which are not available in audio content. An initial step towards achieving this could be to segment the audio into speech-silencemusic.

Finally, it is important to thoroughly evaluate these services to gather more feedback from users. Because we operate on a business-to-business (B2B) model, our clients are mostly TV, radio operators and other media producers; we do not have direct access to their content consumption usage. This makes it challenging to measure the impact of our algorithms on their product. On the contrary companies like Netflix and Spotify, to name a few, when they want to evaluate a new ML model they perform controlled experiments on a random portion of their consumers (A/B testing) to decide whether the new feature increases the content consumption or not [6, 13] and if it does they ship the new feature to production. Currently, most of the use cases described in this paper are only tested in a lab environment with fairly technical users. We plan to serve not only tech-savvy people but want to change the way all users find their content and improve their user experience.

REFERENCES

[1] Ericsson ConsumerLab, “TV and Media 2017, A consumer-driven future of media” (2017). Available at: https://www.ericsson.com/en/trends-and-insights/consumerlab/consumerinsights/reports/tv-and-media-2017

[2] X. Naturel and S. A. Berrani, "Content-Based TV Stream Analysis Techniques toward Building a Catch-Up TV Service," 2009 IEEE ISM.

[3] Tivo, “Content discovery is still too hard - this is how you make it easy” (2017).

[4] MIT Technology Review (2018). 10 breakthrough technologies 2018. Available online at: https://www.technologyreview.com/lists/technologies/2018/

[5] How youtube perfected the feed (2018). Available at: https://www.theverge.com/2017/8/30/16222850/youtube-google-brain-algorithm-videorecommendation-personalized-feed

[6] Recommending music on Spotify with deep learning (2014). Available at: http://benanne.github.io/2014/08/05/spotify-cnns.html

[7] V. D. Oord et al. “Deep content-based music recommendation” 2013 NIPS.

[8] Carlos A. Gomez-Uribe and Neil Hunt. 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. ACM Trans. Manage. Inf. Syst. 2015.

[9] Netflix Technology Blog (2017). Artwork Personalization at Netflix. Available at: https://medium.com/netflix-techblog/artwork-personalization-c589f074ad76

[10] B. Prihasto et al., "A survey of deep face recognition in the wild," 2016 ICOT.

[11] F. Schroff, D. Kalenichenko and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering”, 2015 IEEE CVPR.

[12] Wen et al. “A Discriminative Feature Learning Approach for Deep Face Recognition”, 2016 ECCV.

[13] G. Yandong et al. “MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition”, 2016 arXiv:1607.08221

[14] S. Ren et al. “Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks”, 2015 NIPS

[15] D. Jia et al. “Imagenet: A large-scale hierarchical image database”, 2009 CVPR

[16] K. Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", 2014 arXiv:1406.1078

[17] O. Vinyals et al. “Grammar as a foreign language”. 2015 NIPS.

[18] M. Jaderberg et al. “Reading text in the wild with convolutional neural networks”. 2014, arXiv:1412.5903.

[19] W. K et al. “End-to-end scene text detection”, 2011 ICCV.

[20] S.M. Lucas et al. “ICDAR 2003 robust reading competitions”, 2003 ICDAR.

[21] A. Mishra et al. “Scene text recognition using higher order language priors”. 2012 BMVC.

[22] L. Gomez and D. Karatzas, “Textproposals: a text-specific selective search algorithm for word spotting in the wild”. Pattern Recognition, 2017.

[23] S. Karaoglou et al. “Words matter: scene text for image classification and retrieval”. IEEE transactions on Multimedia, 2017.

[24] Chen-Yu Le and Simon Osindero, “Recursive recurrent nets with attention modeling for ocr in the wild”. 2016 CVPR.

[25] C. Zhanzhan et al. “Focusing attention: Towards accurate text recognition in natural images”. 2017 IEEE ICCV.

[26] Prins M. and Wit J. “Scanning News Videos With An Interactive Filmstrip”. In Adjunct Publication of the 2017 ACM TVX.

[27] RTL Nieuws en TNO personaliseren bulletins (2013). Available at: https://www.svdj.nl/nieuws/rtl-nieuws-en-tno-personaliseren-bulletins/

[28] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE CVPR.

[29] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. 2015 BMVC.

ACKNOWLEDGEMENTS

This work was supported in part by the "MIT-R&D Samenwerkingsproject" (Ref. MTRLA16046). We would also like to thank R. R. Amirkhan and Maurits Bleeker for their valuable feedback.