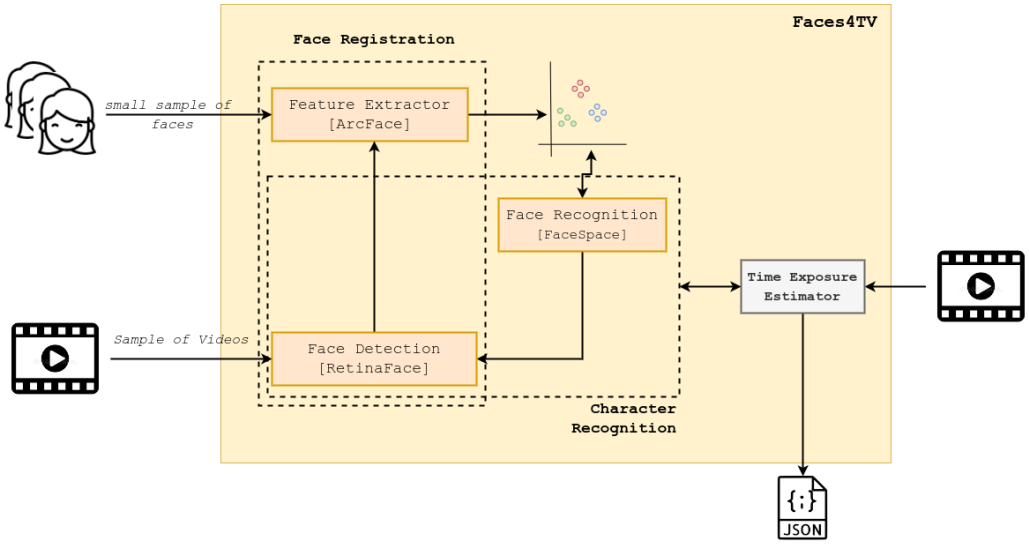
Graphical Abstract

Faces4TV: An Efficient Model for Predicting Character Exposure Time in TV Media

Jose M. Saavedra, Lukas Pavez, Cristobal Loyola, Rodrigo Lara, Juan Carlos Aguirre, Carla Vairetti



Highlights

Faces4TV: An Efficient Model for Predicting Character Exposure Time in TV Media

Jose M. Saavedra, Lukas Pavez, Cristobal Loyola, Rodrigo Lara, Juan Carlos Aguirre, Carla Vairetti

- We present an efficient approach for predicting character exposure time in TV media.
- We present two algorithms, Hysteresis and Continuity, to improve face recognition in TV videos.
- Our approach is dynamic; it can quickly adapt to new characters, a critical property in TV media.
- Our approach deals with difficult face poses appearing in TV videos.
- We also present a new evaluation dataset for TV character recognition with 30 identities.

Faces4TV: An Efficient Model for Predicting Character Exposure Time in TV Media

Jose M. Saavedra[a], Lukas Pavez[a], Cristobal Loyola[a], Rodrigo Lara[b], Juan Carlos Aguirre[b], Carla Vairetti[a]

[a] *Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Chile, Mons. Alvaro del Portillo 23455, Santiago, 7550000, RM, Chile*
[b] *Canal 13, Inés Matte Urrejola 0848, Santiago, 7520285, RM, Chile*

Abstract

Identifying characters in TV media is a critical task for TV broadcasters. This is a special case of the well-known face recognition problem of computer vision. However, the dynamic set of characters, the high variability of faces, and the required precision make the SOTA face recognition solutions not directly appropriate for this problem. Therefore, we present Faces4TV, a recognition model to estimate the exposure time of characters in TV, solving the challenges described above. Our model does not require any training process, and it leverages the face space inferred by the ArcFace model. Thus, our model is dynamic and achieves a high true positive rate with 0 false positives. In addition, to improve the baseline results, we propose two strategies, Hysteresis and Continuity, that allow us to increase the performance from 84.0% to 91.4% on average.

Keywords: face recognition, exposure time, face feature spaces
2000 MSC: 68T10, 68U10, 68U99

1. Introduction

Television broadcasting is the industry that provides television programs around the world. This industry is also a profitable market with sustainable growth in recent years. Indeed, a recent report from The Business Research Company[1] shows that this market increased its revenue from US $ 249 billion

[1] https://www.thebusinessresearchcompany.com/report/television-broadcasting-global-market-report

in 2021 to $ 267 billion in 2022. In addition, the same report establishes that many TV studios are increasing their budgets for TV shows to produce high-quality content.

The television industry is constantly evolving and changing due to new technologies. In fact, the connected TV trend and the competition for delivering new content and advertising make TV broadcasters face new challenges. For instance, TV industries must analyze consumer preferences in finest detail, optimize their content grid, choose their television figures carefully, and define their advertising spaces and associated rates based on "hard data". In addition, traditional broadcasters, like those with a regional scope, rely on a simple business model based on advertising. Thus, to produce effective ads, it is crucial to measure the audience to determine which ads to place at which times and the involved costs.

In this vein, it is crucial to automatically extract information from TV content to correlate it with ratings during different periods. For instance, it is essential for TV broadcasters to accurately know the exposure time of characters like TV presenters, politicians or any influencers or opinion leaders. Character exposure time can be leveraged for many purposes, like understanding the relationship between characters and ratings or labeling the media content to facilitate retrieval.

Identifying characters in TV media falls into the well-known face recognition problem of computer vision. However, there are special differences that make TV character identification a challenging problem. These problems can be summarized as follow:

- **Dynamic Behavior**: the set of characters to be identified is not fixed. It can change dynamically in a short period. Herefore, a traditional supervised model trained over a set of identities can not be directly applied. Furthermore, the supervised strategies rely on a huge amount of labeled data, which become unfeasible in a dynamic environment.

- **High-Variability Poses**: the faces in TV media can be affected by various noises, making the poses undergo high variations. For instance, faces can appear in different scales depending on the TV camera settings, or they can show variations concerning the presence of objects like beards, mustaches, glasses or mouth masks. In addition, TV can contain characters in the wild, like reporters in the news scene.

- **High-Precision of Exposure Time**: precision is highly relevant for

TV managers to make accurate decisions. The goal is to achieve a more precise character exposure time even though the faces undergo variability, occlusion or the characters show a dynamic appearance in the TV media.

On the other hand, the explosion of deep learning has allowed us to see enormous advances in research and applications of artificial intelligence, powering areas like machine learning, computer vision and natural language processing. In the case of computer vision, we have seen outstanding results in image classification [1, 2], object detection [3, 4, 5], image segmentation [6, 7, 8], and image synthesis [9, 10]. We also have seen surprising results in text generation [11] and speech recognition [12]. Face recognition is one of the most popular and widely used computer vision tasks. This task aims to identify a person through their face image. Commonly, face recognition goes after a face detection stage to separate the relevant information (the face) from the rest of the image. Deep learning-based models, through convolutional neural networks, have contributed to increasing the performance of face detection and recognition significantly [13].

In the context of computer vision, the effectiveness of a model for a specific task relies directly on the quality of features extracted from the images. If we can extract semantic and discriminative features from the interest object, the performance on specific tasks like classification, segmentation, and detection, among others, will increase. In the context of face recognition, ArcFace [14] is an effective neural network for computing discriminative features from faces. The ArcFace's learning strategy is based on generating face representations to increase intra-class compactness and inter-class discrepancy. This characteristic makes it appropriate in problems where the number of classes is not fixed, even though this model does not overcome the challenges described previously.

Therefore in this work, we present a methodology called Faces4TV to precisely compute the exposure time of characters in TV media, tackling the dynamic behavior of characters to be identified, the high variability of face poses and the required precision for exposure time estimation. Faces4TV is based on the feature space inferred by ArcFace [14] and incorporates two new algorithms, Hysteresis and Continuity, that leverage the already recognized face to improve recognition of difficult faces of a video shot.

The paper is organized as follows: Section 2 describes the related works, Section 3 explains our proposal in detail, Section 4 describes the experimental

setting and the achieved results. Finally, Section 5 presents the conclusions.

2. Related Work

Face recognition is one of the most popular tasks in computer vision, widely used in a diversity of areas. It is also one of the longest-standing research topics in artificial intelligence and computer vision [15]. The advent of deep-learning-based models has facilitated the proliferation of a diversity of applications by releasing not only high-precision models but also large face datasets [16].

The state-of-the-art architecture for face recognition relies on *siamese networks* that share a CNN backbone between two branches to generate highly-discriminative representations from face images. Pioneers using CNN for learning face representations are DeepFace [17], and VGGFace [18], where the last introduce the triplet loss for face recognition. One popular face collection for testing in this domain is Labeled Faces in the Wild (LFW), where DeepFace showed an accuracy of 97.35% and VGGFace 98.95%.

More recently, Deng et al. proposed ArcFace [14], achieving an accuracy of 99.5% in the LFW dataset. ArcFace proposes the *Additive Angular Margin Loss* aiming to increase the distance between representations of different classes, which reduces the number of false positives. This proposal also produces a face space where representations of the same face are pulled to form more compact groups. Moreover, instead of using a distance loss, ArcFace utilizes the angle between vectors to represent their differences. Another important feature of ArcFace is its strategy to deal with noisy data. To deal with this problem, ArcFace uses sub-center representations for different classes that are then compared with an input representation. Figure 1 shows the scheme of the sub-center strategy proposed in ArcFace.

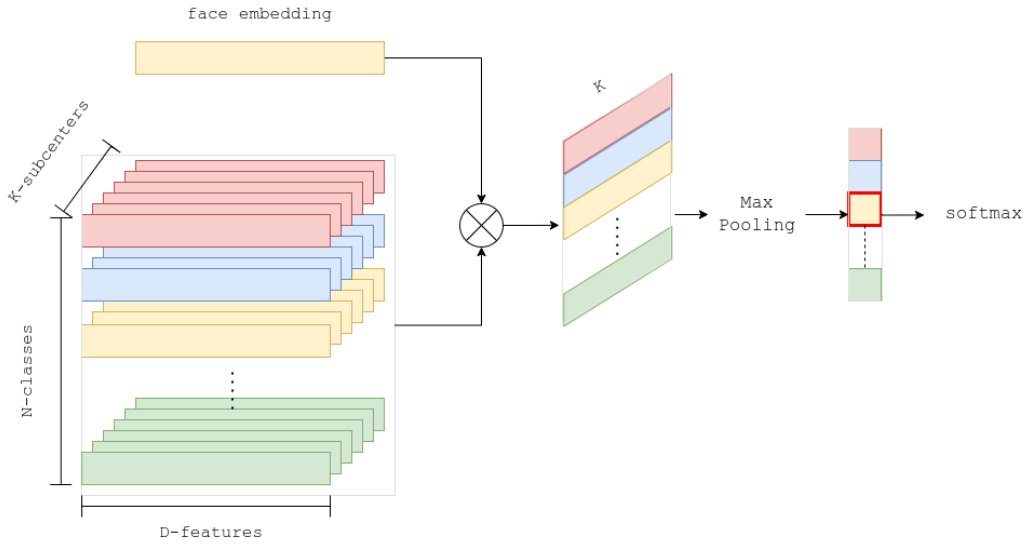


Figure 1: Scheme of the subcenter-based strategy applied by ArcFace.

A face recognition approach assumes the underlying model has a face as input. However in real applications, we instead have a complete crowded image as input. Thus, a robust face detection method is a critical stage. In the literature, RetinaFace [19] provides accurate and efficient face detection in the wild. RetinaFace is trained using multitask loss as described in Equation 1.

$$\begin{aligned} \mathcal{L} = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) \\ + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel} \end{aligned} \tag{1}$$

where L_{cls} and L_{box} are the traditional losses used for classification and regression [20, 8]. RetinaFace proposes two additional losses, a landmark regression loss L_{pts} and a mesh decoder loss L_{pixel}. The *landmark regression* is inspired by MaskRCNN [8] that achieved improvements merging pixel-wise segmentation with bounding box prediction. In this vein, RetinaNet showed that predicting five facial points in a face improves the performance of detection. Finally, the *mesh decoder loss* improves detection by forcing latent representation to be good enough to produce a 3D model of the face. For the last case, RetinaNet leverages the results of Zhou et al. [21] for Dense 3D face decoding.

2.1. Challenges in TV Media

As commented previously, the exposure time of characters in TV is an important task for TV broadcasters. The time a character appears on TV can be reduced to a problem of character recognition in a video. Although we already have outstanding models for face recognition, the problem we focus on still has the following challenges:

1. The number of characters to recognize presents high variability. We can not know in advance which characters will be required to analyze. These will be added or removed dynamically.

2. The characters' faces can appear in various poses or are occluded with some accessories that make recognition difficult. Figure 2 shows examples of challenging faces we need to deal with.

3. The precision rate of recognition is highly critical to allow managers to make good decisions.

Figure 2: Examples of difficult faces appearing in TV media.

Therefore, in this work we present a methodology to estimate exposure time of characters in TV, solving the challenges described above. Thus, our model is dynamic and achieves high true positive rate (around 91.4% in average) with 0 false positives. Our model does not require any training process, and it leverages the face space inferred by the ArcFace model. In addition, to improve the baselines results, we propose two strategies that allow us to increase the performance from 84.0% to 91.4% in average.

3. Faces4TV

Our proposal aims to accurately estimate characters' exposure time in TV media. To this end, our proposal focuses on improving recognition of hard

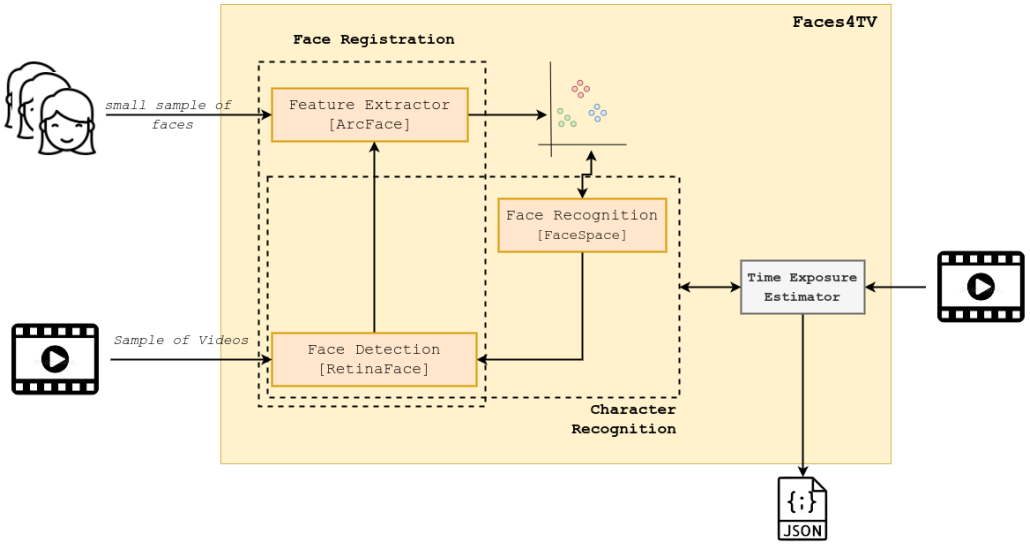


Figure 3: General scheme of our proposal Faces4TV. We highlight the three main process: Face Registration, Character Recognition and Time Exposure Estimator.

faces by leveraging information from the own video shot. Figure 3 illustrates the big picture of our proposal, which is composed of three main processes: face registration, character recognition and exposure time estimation. These processes are described in detail in the following paragraphs.

3.1. Face Registration

This is the first stage of our proposal. Its goal is to generate an effective face space, where face embeddings from the same character fall close together, and embeddings of different characters fall apart from each other. In this proposal, we use ArcFace [14] as the feature extractor model to produce a face space. For clarity, we define $\mathcal{E} : image \rightarrow R^d$ as the ArcFace feature extractor that receives an image and produce a d-dimensional feature vector (the embedding). Figure 4 shows the face space produced by ArcFace for around 25 instances of 10 different characters. We observe how ArcFace can successfully keep different characters separated from each other and maintain close face representations of the same identity.

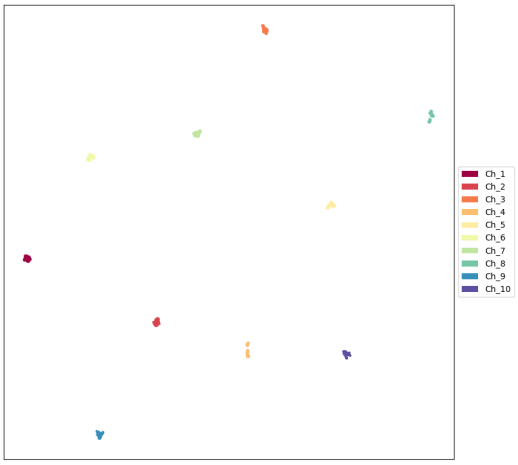


Figure 4: UMAP visualization of the face space produced by ArcFace for 10 different characters.

The face registration process is triggered when new characters need to be enrolled. When this happens, we apply the following stages:

1. **Initial Registration**: when a new character κ requires to be enrolled, our proposal needs a small set of face examples ($\sim$ 15 is suggested) of the new character κ. These images are passed through the ArcFace feature extractor to get the corresponding embeddings. We call Ω_κ to this initial set of embeddings for κ. Ω_κ is then saved along with the corresponding character's identity.

2. **Boosted Registration**: having an initial set of embeddings Ω_κ for a character κ, we will extend it automatically using a set of video shots where the underlying character appears. Unlike the initial registration process, we do not have isolated face images as input; we now have a set of videos. Therefore, we use the RetinaFace [19] model to extract faces from the input video that are then passed through the feature extractor to produce a set of face embeddings Ψ. Thus, we update the initial embedding set Ω_κ by adding those embeddings $e \in \Psi$ that are close to μ_κ. Here, μ_κ is the mean of Ω_κ. The closeness is measured by

the Euclidean distance and a threshold th. Equation 2 express how we extend the initial embedding set Ω_κ.

$$\Omega_\kappa = \Omega_\kappa \bigcup \{e \in \Psi | ||e - \mu_\kappa||_2 < th\} \tag{2}$$

3.2. Character Recognition

This is the core of our proposal due to the exposure time relies on accurate character recognition.

The character recognition process starts defining K face embedding centers representing the variability of a character. To this end, we form K clusters for all the embeddings representing each character. In this manner, each character is represented by K embeddings. For clarity, we define $\bar{\Omega}_\kappa$ as the set of centers for a character κ. Thus, $\bar{\Omega}^k_\kappa$ represents the k-th center.

With the face space represented by Ω_κ, we are ready to detect the occurrence of a character in an input video. The general idea is to detect faces in the input video and then compare each detected face with each embedding center of $\bar{\Omega}_\kappa$, for each character κ.

For the recognition process, we define a similarity function $sim_\kappa : \mathcal{R}^d \rightarrow \mathcal{R}$, that receives an embedding and returns a similarity score with respect to the identity κ. Formally, we define sim_κ as follows:

$$sim_\kappa(e) : \max_{\mu \in \bar{\Omega}_\kappa} \frac{e \cdot \mu^T}{||e||||\mu||} \tag{3}$$

Therefore, a face A is recognized as character κ if $sim_\kappa(\mathcal{E}(A)) \geq th_r$ and $\forall_{\kappa' \neq \kappa} sim_{\kappa'}(\mathcal{E}(A)) < sim_\kappa(\mathcal{E}(A))$. Here, th_r is the recognition threshold.

However, in TV, we need to deal with complex faces like those with a difficult pose or those having extra accessories that occlude part of the face. This situation represents the main challenge we seek to solve through this proposal.

Our solution leverages an important property of faces in a video. A face of a character does not change abruptly from frame to frame. There is a soft movement between a straight face to a rotated face (profile face). This good situation motivated us to propose the following mechanisms to improve the recognition rate:

1. **Hysteresis**: this idea was inspired by the technique of the same name applied by the well-known Canny method [22]. The idea is to propagate

the recognition of a character with a high score in a frame at time t_i to the subsequent frames. There is a high probability of finding a character after an occurrence is detected with a high score. The implementation of hysteresis would consist on relaxing the recognition threshold at time t_{i+1} for a character κ if a face of κ was detected with a high score in t_i. To this end, we use a lower recognition threshold th_{low}. In this manner, our model can recognize hard faces if they are close to easier faces. This situation is very similar to the strong and weak edges described by Canny [22], where weak edges become strong if they are connected with any strong edge. In our case, a weakly recognized face becomes strongly recognized if the first follows the second. Figure 5 illustrates the hysteresis mechanism, where the three first faces are strongly recognized ($score > th_r$), and the last two are not because their score is below th_r. However, as they follow a strong recognition and have a score greater or equal to th_{low}, they become recognized faces with the same identity as the three first.

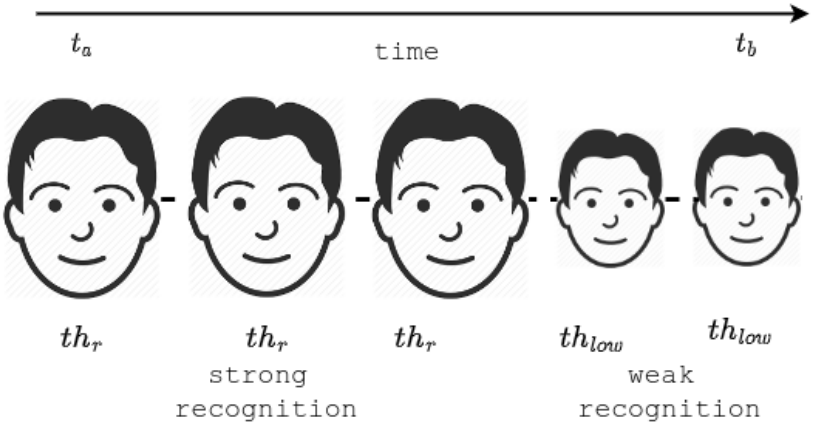


Figure 5: Scheme of the proposed hysteresis mechanism for face recognition.

2. **Continuity**: in this case, we exploit the property of continuity of a face along with a TV video. This property establishes that if we have a face A recognized as character κ in time t_i, there will be another face B of κ in t_{i+1}, under the following two conditions:

   2.1. Appearance Similarity: $||\mathcal{E}(A) - \mathcal{E}(B)||_2 \leq th_{sim}$

   2.2. Locality: $||pos(A) - pos(B)||_2 \leq th_{pos}$

where $\mathcal{E}(\cdot)$ is the embedding function and $pos(\cdot)$ returns the center position of a detected face. In addition, th_{sim} and th_{pos} are thresholds for appearance similarity and locality, respectively.

Therefore, given a face recognized as κ at t_i, our continuity mechanism assesses any detected face in t_{i+1} that did not pass the hysteresis evaluation. If a detected face matches the two above conditions, that face is recognized as κ. Figure 6 illustrates our proposal.

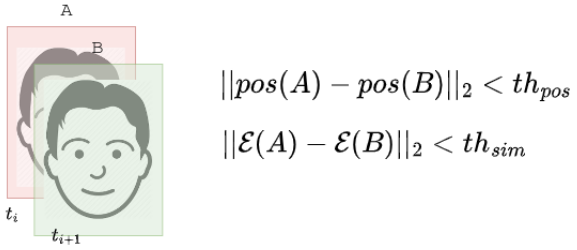


Figure 6: Scheme of the Continuity strategy.

3.3. Exposure Time Estimation

This is the ultimate goal of our proposal. Given the information produced by the two processes discussed previously, we infer the exposure time of a character in a TV video. First, we define an occurrence of a character as a sequence of frames where the character is recognized, with the condition that the number of frames between each recognition is less than the video's FPS, as shown in Figure 7. With this condition, the character has to disappear for 1 second to determine the end of the occurrence. The time of an occurrence is calculated with the first and last frame of the occurrence with the equation 4. Finally, we define the total exposure time of a character as the sum of all the occurrence times of that character.

$$time_{occ} = \frac{frame_n - frame_1}{fps} \tag{4}$$

4. Experiments and Discussion

In this section, we describe the experimental protocol and the achieved results and discuss the achievements. We start by defining the used datasets and metrics. Then we present a strategy to deal with a huge amount of face

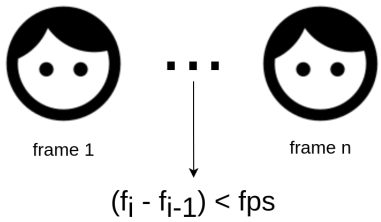


Figure 7: Sequence of frames that define the occurrence of a character.

Figure 8: A collage showing a sample of characters defined for our proposal evaluation.

embeddings for each character and describe the role of our two proposals: hysteresis and continuity.

4.1. Datasets

We describe the datasets used in our experiments regarding three aspects: the number of characters to be analyzed, the TV videos used in the registration process and the testing videos used to compare our results. We describe each of the mentioned aspects in the following lines:

1. **Characters**: we define 30 characters from a regional TV broadcaster. The occurrences of the characters present high variability concerning scale, occlusion, and poses, among other variations. Figure 8 present a collage with a sample of the defined characters.

2. **Videos for the Registration Process**: we collected 70 TV videos containing the characters defined previously. These videos are extracted from TV programs, like news, TV shows and soap operas, with

different duration, from 2 minutes to 1 hour. We will call *FaceSpace Videos* to these videos.

We then compute the embeddings for all the characters' faces appearing in the FaceSpace Videos. This process produces a large number of embedding for characters. Table 1 shows the number of embeddings for each character in the face space.

Character	Quantity	Character	Quantity	Character	Quantity
Ch_1	69552	Ch_{11}	29012	Ch_{21}	20625
Ch_2	28876	Ch_{12}	13460	Ch_{22}	4750
Ch_3	14111	Ch_{13}	39287	Ch_{23}	16655
Ch_4	1413	Ch_{14}	18520	Ch_{24}	62122
Ch_5	38494	Ch_{15}	35139	Ch_{25}	10715
Ch_6	72154	Ch_{16}	81580	Ch_{26}	17600
Ch_7	11338	Ch_{17}	12095	Ch_{27}	20850
Ch_8	27518	Ch_{18}	6285	Ch_{28}	4195
Ch_9	10205	Ch_{19}	8270	Ch_{29}	2295
Ch_{10}	33397	Ch_{20}	9000	Ch_{30}	6345

Table 1: Quantity of embeddings for all characters.

3. **Testing Videos**: During our experiments, we set 20 videos of 10 minutes each and labeled them every 30 frames (1 second). Consequently, we get a sample of 600 frames per video, where most appeared with one or more characters.

4.2. Metrics

As the exposure time directly depends on the accuracy of the character recognition, we will use metrics related to face recognition to quantify the performance of our proposal:

- **True Positive Rate (TP)**: this metric determines the correct recognition rate of a method. It is computed as the number of correctly recognized faces among the total faces for a given character. Thus, TP ranges from 0 to 1, where 1 is the optimal result.

- **False Positives**: this is an absolute number, indicating the number of faces incorrectly recognized as a specific character. The optimal value is 0, showing that there are no mislabeled faces.

4.3. Clustering

The large size of embeddings for each character shown in Table 1 can affect the efficiency of the proposal. In addition, it is highly probable to produce redundant embeddings. To alleviate these problems, we propose to cluster each character's embeddings into only K representations using the well-known k-means algorithm. This solution also deals with the diversity of faces for each character.

Specifically, we propose three clustering variations:

- **Mean**: the most straightforward strategy is to collapse all the embeddings of a character into one embedding computed as the mean of all of them.
- **5-means**: this represents a small number of centers for each character.
- **20-means**: this represents a large number of centers for each character.

It is important to mention that we also have run experiments with more than 20 centers for each character, but we did not observe improvements. In addition, it is also crucial to keep the number of centers low to maintain the efficiency of the proposal.

4.4. Parameter Setting

In all our experiments, we set $th_r = 0.5$, $th_{sim} = 0.3$ and $th_{pos} = 15$.

4.5. Results

The first step is to determine the clustering strategy to be used. To this end, we conducted experiments in videos containing all characters. Table 2 shows the performance of the baseline model using the three strategies of clustering described in Section 4.3. We can note the superiority of 20-Means over the other two, showing an improvement of 8.2% over the results achieved by the Mean strategy and 1.8% over 5-Means.

4.5.1. Hysteresis and Continuity

Choosing 20-Means as the clustering strategy for grouping embeddings for each character, we then evaluate the modules of hysteresis and continuity aiming to improve the baseline results. With respect to hysteresis, we tested two low recognition thresholds: 0.15 and 0.30. We run hysteresis experiments with and without continuity. Table 3 summarize these results.

	Mean		5-means		20-means	
Character	TP	FP	TP	FP	TP	FP
Ch_1	0	0	75.37	0	**76.87**	0
Ch_2	89.47	0	**94.74**	0	**94.74**	0
Ch_3	60.00	0	**80.00**	0	**80.00**	0
Ch_4	82.49	0	85.52	0	**87.54**	0
Ch_5	73.33	0	**80.00**	0	73.33	0
Ch_6	68.75	0	72.80	0	**76.60**	0
Ch_7	84.86	0	86.59	0	**88.17**	0
Ch_8	71.27	0	71.64	0	**72.00**	0
Ch_9	75.07	0	75.34	0	**75.88**	0
Ch_{10}	88.29	0	89.76	0	**95.12**	0
Ch_{11}	26.73	0	46.08	0	**48.85**	0
Ch_{12}	65.59	0	65.59	0	**69.89**	0
Ch_{13}	75.96	0	80.92	0	**85.14**	0
Ch_{14}	85.57	0	**87.63**	0	**87.63**	0
Ch_{15}	56.02	0	65.97	0	**71.99**	0
Ch_{16}	86.05	0	90.23	0	**92.09**	0
Ch_{17}	**100**	0	**100**	0	**100**	0
Ch_{18}	81.21	0	82.55	0	**84.56**	0
Ch_{19}	88.00	0	91.20	0	**93.60**	0
Ch_{20}	**94.40**	0	**94.40**	0	**94.40**	0
Ch_{21}	83.87	0	86.29	0	**89.52**	0
Ch_{22}	70.00	0	70.00	0	**72.00**	0
Ch_{23}	72.65	0	76.07	0	**83.76**	0
Ch_{24}	60.26	0	**69.23**	0	**69.23**	0
Ch_{25}	71.01	0	78.26	0	**82.61**	0
Ch_{26}	96.55	0	97.81	0	**98.12**	0
Ch_{27}	88.68	0	**92.45**	0	**92.45**	0
Ch_{28}	88.78	0	90.82	0	**91.84**	0
Ch_{29}	94.74	0	97.37	0	**98.68**	0
Ch_{30}	93.71	0	92.31	0	**94.41**	0
Avg	75.80	0	82.20	0	**84.00**	0

Table 2: Impact of the clustering strategy (Mean, 5-Means. 20-Means) for character recognition. We show the results in terms of true positive rate (TP) and false positive (FP).

As expected, a lower threshold tends to increase the false positives, as shown in Table 3. Here, $th_{low} = 0.15$ produces 6.3 false positives on average, with cases in which the false positives reach 25 or 50. However, increasing the value of th_{low} to 0.30 allows us to improve the true positive rate, keeping the number of false positives at 0. We tested other values for th_{low} less than 0.30, but the lower one with $FP = 0$ was 0.30.

From Table 3, we can also observe the benefit of using continuity after hysteresis. It follows that continuity keeps the performance of hysteresis or

	base model		$hist_{0.15}$		$hist_{0.15}$+cont		$hist_{0.3}$		$hist_{0.3}$+cont		cont	
Chr	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
Ch_1	76.87	0	**95.52**	0	**95.52**	0	88.06	0	**95.52**	0	**95.52**	0
Ch_2	94.74	0	**97.37**	1	**97.37**	1	**97.37**	0	**97.37**	0	**97.37**	0
Ch_3	**80.0**	0	**80.00**	0	**80.00**	0	**80.00**	0	**80.00**	0	**80.00**	0
Ch_4	87.54	0	**94.95**	1	**94.95**	1	94.28	0	**94.95**	0	**94.95**	0
Ch_5	73.33	0	**86.67**	0	**86.67**	0	**86.67**	0	**86.67**	0	80.0%	0
Ch_6	76.60	0	**90.12**	11	**90.12**	11	86.57	0	87.25	0	83.70	0
Ch_7	88.17	0	**94.79**	11	**94.79**	11	93.53	0	93.69	0	91.80	0
Ch_8	72.00	0	**84.36**	3	**84.36**	3	79.27	0	80.00	0	84.00	0
Ch_9	75.88	0	**88.08**	2	**88.08**	2	86.72	0	87.80	0	82.66	0
Ch_{10}	95.12	0	**100.0**	25	**100.0**	25	99.51	0	**100.0**	0	**100.0**	0
Ch_{11}	48.85	0	**97.24**	50	**97.24**	50	88.02	0	90.78	0	96.77%	0
Ch_{12}	69.89	0	**78.49**	0	**78.49**	0	75.27	0	**78.49**	0	**78.49**	0
Ch_{13}	85.14	0	**95.41**	13	**95.41**	13	92.48	0	93.30	0	89.17	0
Ch_{14}	87.63	0	**91.75**	4	**91.75**	4	89.69	0	89.69	0	89.69	0
Ch_{15}	71.99	0	**87.17**	19	**87.17**	19	83.64	0	84.29	0	84.95	0
Ch_{16}	92.09	0	**98.14**	20	**98.14**	20	97.67	0	**98.14**	0	**98.14**	0
Ch_{17}	**100.0**	0	**100.0**	1	**100.0**	1	**100.0**	0	**100.0**	0	**100.0**	0
Ch_{18}	84.56	0	**93.96**	0	**93.96**	0	90.60	0	91.95	0	92.62	0
Ch_{19}	93.60	0	**96.80**	4	**96.80**	5	95.20	0	96.00	0	**96.80**	0
Ch_{20}	94.40	0	**96.80**	0	**96.80**	0	95.20	0	**96.80**	0	96.00	0
Ch_{21}	89.520	0	**96.77**	0	**96.77**	0	94.35	0	94.35	0	95.16	0
Ch_{22}	72.00	0	80.00	7	80.00	7	80.00	0	**82.00**	0	78.00	0
Ch_{23}	83.76	0	**93.16**	1	**93.16**	1	90.60	0	92.31	0	**93.16**	0
Ch_{24}	69.23	0	**76.92**	1	**76.92**	1	75.64	0	**76.92**	0	**76.92**	0
Ch_{25}	82.61	0	**85.51**	1	**85.51**	1	**85.51**	0	**85.51**	0	**85.51**	0
Ch_{26}	98.12	0	**98.43**	6	**98.43**	6	98.12	0	**98.43**	0	**98.43**	0
Ch_{27}	92.45	0	**98.11**	1	**98.11**	1	96.6	0	97.36	0	97.74	0
Ch_{28}	91.84	0	**96.94**	6	**96.94**	7	95.92	0	**96.94**	0	95.92	0
Ch_{29}	**98.68**	0	**98.68**	2	**98.68**	2	**98.68**	0	**98.68**	0	**98.68**	0
Ch_{30}	94.41	0	**95.80**	0	**95.80**	0	**95.80**	0	**95.80**	0	**95.80**	0
Avg	84.00	0	**92.30**	6.3	**92.30**	6.4	90.40	0	**91.40**	**0**	90.90	0

Table 3: Impact of hysteresis and continuity for character recognition. We present the results in terms of true positive rate (TP) and false positive (FP).

improves it. For instance, for Ch_1 we improve the TP metric from 86.06%, achieved by the use of hysteresis ($th = 0.30$), to 95.52% after applying continuity.

In addition, we also observe the complementary behavior of using both hysteresis and continuity. When we use each proposal separately, the performance is lower than when both are applied together.

4.6. Qualitative Results

4.6.1. Hysteresis and Continuity

Figures 9 and 10 show examples of outstanding performance of our proposal over the baseline. The first row of each figure shows how the base model fails to recognize almost all faces. This is because the faces correspond to hard cases. When we add hysteresis, we can recognize most of them. We can see this behavior in the second row for both figures. The base model fails because the recognition scores are below 0.5, but when we use hysteresis with $th_{low} = 0.3$ the model can recognize most of the harder cases. Finally, when we add continuity (third row), we can recognize faces with a score below the lower threshold because they move softly in the frame sequence.

4.6.2. Hard Faces

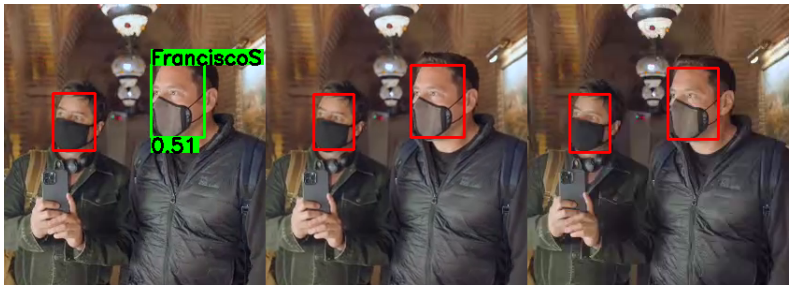
Through our proposal, we improve the true positive rate. However, we have situations like the ones shown in Figure 11, where a face appears occluded, rotated or in a very small scale.

However, the false negatives produced by hard faces do not significantly impact the estimation of total exposure time because the non-recognized faces represent a tiny portion of the whole sequence. For example, Figure 12 shows a false negative, where the person is wearing a hat and has his eyes closed. Analyzing the sequence of frames where the false negative appears (see Figure 13), we observe that most of the frames are recognized successfully (11 of 12). In general, the error produced by our proposal is below 1 second, which is a negligible time for the TV industry.

5. Conclusions

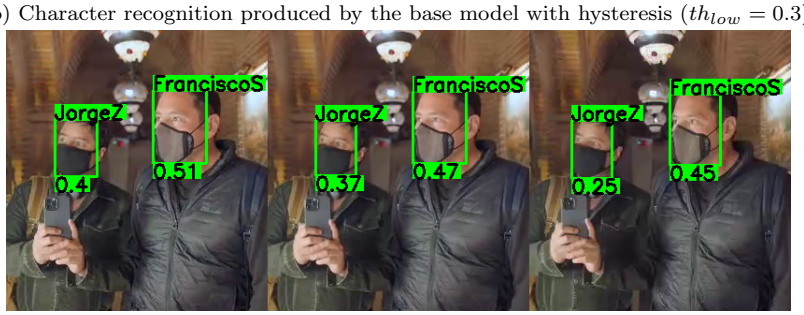
We present Faces4TV, a model to accurately estimate the exposure time of characters appearing in TV videos. Our approach leverages ArcFace to build a face space over which we perform a similarity based strategy to recognize characters. To deal with different noise appearing in the videos we propose Hysteresis and Continuity that showed to improve the TP in 7.4%, keeping 0 false positives. In addition, our proposal is dynamic in that it can be easily adapted to any target characters and is robust to difficult faces.

Therefore, our model represents an excellent tool for TV broadcasters, allowing managers to make better decisions. Moreover, beyond the exposure time, the TV industry can leverage our proposal in other tasks like the



(a) Character recognition produced by the base model, which does not include hysteresis nor continuity.

(b) Character recognition produced by the base model with hysteresis ($th_{low} = 0.3$).



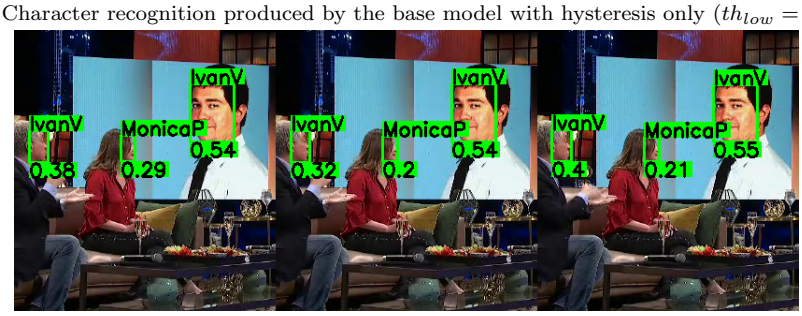
(c) Character recognition produced by the base model with hysteresis and continuity ($th_{low} = 0.3$).

Figure 9: Qualitative results comparing the base model with our proposal based on hysteresis and continuity.

evaluation of genre equality of characters or determining the perception of characters by viewership.

(a) Character recognition produced by the base model, which does not include hysteresis nor continuity.

(b) Character recognition produced by the base model with hysteresis only ($th_{low} = 0.3$).

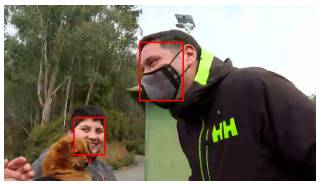


(c) Character recognition produced by the base model with hysteresis and continuity ($th_{low} = 0.3$).

Figure 10: Qualitative results comparing the base model with our proposal based on hysteresis and continuity.

(a) Occluded example.

(b) Rotated example.



(c) Mask example.

Figure 11: Examples of hard faces.

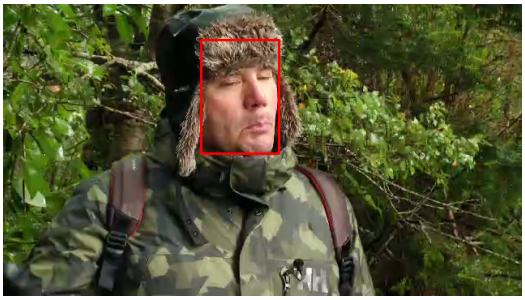


Figure 12: An example of a false negative.

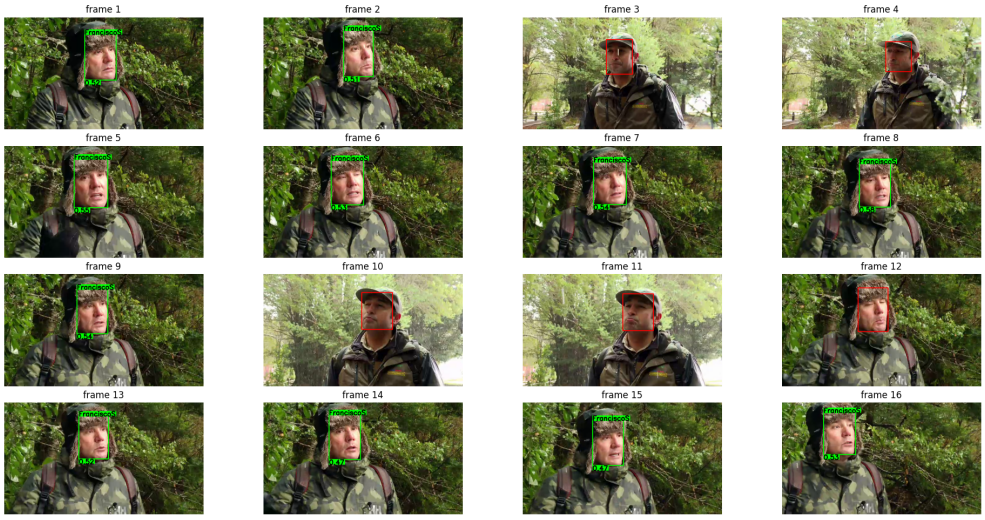


Figure 13: Sequence of frames for Figure 12.

References

N. Jmour, S. Zayen, A. Abdelkrim, Convolutional neural networks for image classification, in: 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2018, pp. 397–402. doi:10.1109/ASET.2018.8379889.

K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 770–778.

J. Deng, X. Xuan, W. Wang, Z. Li, H. Yao, Z. Wang, A review of research on

object detection based on deep learning, Journal of Physics: Conference Series 1684 (2020) 012028. doi:10.1088/1742-6596/1684/1/012028.

J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 779–788.

M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 6105–6114. URL https://proceedings.mlr.press/v97/tan19a.html

S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (7) (2022) 3523–3542. doi:10.1109/TPAMI.2021.3059968.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 834–848. doi:10.1109/TPAMI.2017.2699184.

K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322.

M. Abdolahnejad, P. Liu, Deep learning for face image synthesis and semantic manipulations: a review and future perspectives, Artificial Intelligence Review 53 (12 2020). doi:10.1007/s10462-020-09835-4.

T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4396–4405. doi:10.1109/CVPR.2019.00453.

S. Zhou, Research on the application of deep learning in text generation, Journal of Physics: Conference Series 1693 (2020) 012060. doi:10.1088/1742-6596/1693/1/012060.

P. Lakkhanawannakun, C. Noyunsan, Speech recognition using deep learning, in: 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2019, pp. 1–4. doi:10.1109/ITC-CSCC.2019.8793338.

S. D. Lionel Landry, E. Fute Tagne, E. Tonye, Cnnsfr: A convolutional neural network system for face detection and recognition, International Journal of Advanced Computer Science and Applications 9 (2018) 240. doi:10.14569/IJACSA.2018.091235.

J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4685–4694. doi:10.1109/CVPR.2019.00482.

I. Masi, Y. Wu, T. Hassner, P. Natarajan, Deep face recognition: A survey, in: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018, pp. 471–478. doi:10.1109/SIBGRAPI.2018.00067.

G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007).

Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708. doi:10.1109/CVPR.2014.220.

O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: British Machine Vision Conference, 2015.

J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou, Retinaface: Single-shot multi-level face localisation in the wild, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5202–5211. doi:10.1109/CVPR42600.2020.00525.

S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1137–1149. doi:10.1109/TPAMI.2016.2577031.

Y. Zhou, J. Deng, I. Kotsia, S. Zafeiriou, Dense 3d face decoding over 2500fps: Joint texture & shape convolutional mesh decoders, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1097–1106.

J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6) (1986) 679–698. doi:10.1109/TPAMI.1986.4767851.