# Scaling up Cast Face Detection in Videos at Globo

**Felipe A. Ferreira**[1,2], **Bruno P. Oliveira**[2], **Rodrigo V. Kassick**[2], **Vinícius Furlan**[2], **Hélio Lopes**[1]

[1]Departamento de Informática – PUC-Rio
Rio de Janeiro, RJ – Brazil

[2]Big Data and AI - Globo
Rio de Janeiro, RJ – Brazil

faferreira@inf.puc-rio.br, iobruno@outlook.com, kassick@gmail.com

vinicius.furlan@g.globo, lopes@inf.puc-rio.br

***Abstract.*** *It has been recognized that a significant increase in the production and consumption of video content occurred in the last decade. Many entertainment companies, like Globo, face challenges regarding video metadata generation. The objective of this paper is to present a suitable architecture for the Globo Group to automatically identify actors that appear in each scene of a video stream, generating new metadata annotations that can be used by recommender systems and search engines among different other applications in this industry sector.*

***Resumo.*** *É reconhecido que ocorreu um aumento significativo na produção e consumo de conteúdo de vídeos na última década. Muitas empresas de entretenimento, como a Globo, enfrentam desafios em relação à geração de metadados de vídeo. O objetivo deste artigo é apresentar uma arquitetura adequada para o Grupo Globo identificar automaticamente os atores que aparecem em cada cena de um stream de vídeo, gerando novas anotações de metadados que podem ser utilizadas por sistemas de recomendação e buscadores entre diferentes outras aplicações do setor.*

## 1. Introduction

In recent years, a significant increase in the production and consumption of video content has been observed. Such content is created by film producers, TV companies, and common users that sometimes are interested in sharing some day-life events.

The Globo Group is considered the world's second-largest TV network. It employs over 12,000 workers and produces an average of 5,500 hours of journalism and entertainment, earning until today 12 Emmy prizes. It retains 36.9% in Brazil's audience share. As a Media tech company, Globo Group distributes both original and third-party content, retaining a large amount of video content in its database. In addition, Globo is constantly developing technologies to overcome challenges in media creation and distribution.

Video content brings a diversity of challenges involving its distribution to users, for instance, search and recommender systems can support users in order to find relevant content inside Globo platforms.

Like many other companies, Globo also faces challenges regarding video metadata. For instance, during content registration, most of a video metadata is not annotated, generating a large number of unlabeled videos. However, one metadata information that is present in most videos is the cast information, with actors' names, roles, and face pictures.

This paper presents a suitable architecture for the Globo Group to automatically identify actors that appear in each scene of a video stream, generating new metadata annotations that can be used by recommender systems and search engines among different products at Globo. Our architecture can operate on large collections of videos and correctly identify actors cast in many different roles.

## 2. Methods

Classical supervised machine learning approaches applied to face identification in images assume the availability of an input image that can be fed to the model, which will output one of many classes. Each of these classes corresponds to some identified person. These approaches work well in contexts where the number of output classes is relatively small and does not change frequently, since every change in the number of classes incurs retraining the model to include the new classes.

In the context of Globo, on the other hand, such changes in the number of the output classes are necessarily frequent, as new actors are cast in diverse roles in soap operas, series and as new productions are added to the inventory. In this scenario, retraining the model in the needed frequency poses serious scalability issues.

To mitigate these issues, the proposed approach follows the idea of [Schroff et al. 2015]. Instead of training a model that takes an image and outputs the classes associated with the individual actors, the processing is separated into two steps: the first is responsible for the identification of faces in the input images; the second is responsible for classifying the detected faces. In the second step, the model extracts embeddings for each detected face and after that compares them with previously computed embeddings extracted from images of the known actors in the cast – a labeled embeddings dataset. The identified faces will be associated with the ones from the labeled dataset which have the smaller distance in the embeddings space.

Following this methodology, initially, we experimented with two approaches to identify the actors in videos, described in the following sections.

### 2.1. One-pass detection

In the one-pass approach (figure 1), we sample one frame per second and then apply the same methods for face detection and embedding extraction as described in [Pena et al. 2020]. Next, we use a pre-trained model [Schroff et al. 2015] to extract embeddings for each detected face. Given these candidate embeddings, the classification process consists of calculating the Euclidean distance of the candidate embedding in relation to the labeled embeddings from the dataset. The output label produced in the classification step will be the one with the shortest distance from the candidate embeddings.

This naive approach where each sampled frame is processed independently,

assumes the image quality as good enough for face recognition algorithms, which, for the matter of processing speed alone, is satisfactory.

In fact, in this approach, the frames are processed just once, and with that, many issues arise, such as failure in accurately recognizing the faces, since they're needed to be in frontal positions, or even false positives, when the face detected corresponds to that of a different casting member.


**Figure 1. High level schema that explains the one pass approach for actors recognition. The steps: Face Detection, Embedding Extraction and Classification are detailed in [Pena et al. 2020]**

## 2.2. Two-pass detection

In an attempt to improve the accuracy and mitigate the issues encountered with the first approach, we've decided to split the video into several scenes first and then process each scene twice, hence, two-pass detection (forward and backward).

In [Pena et al. 2020], we describe each step in details. Right after the scene segmentation, the first pass process the video clip forward and independently, which outputs the frames list and the recognized faces respectively. This is followed by the second pass, which processes the very same clip backward, also outputting the respective frames and faces detected.

Finally, a merge pass compares the results obtained from the two-pass processing, in an attempt to fetch the most accurate results, to solve the issues of undetected faces in the first pass, and also to mitigate false positives that might have occurred.

Figure 2 shows the high level concept of the steps taken:

## 2.3. Current Architecture

The proposed architecture is based on an event-driven streaming platform. It relies on Kafka[Sax 2019] at its core. And it is initiated when a new video file is pushed to a specific bucket on Google Cloud Storage ("raw"), which triggers a Cloud Function that pushes the file metadata to a given Kafka Topic (raw-video-refs).

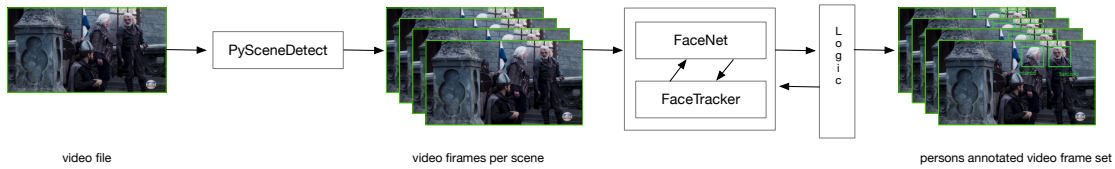


**Figure 2. High level schema that explains the Globo Face Stream system. This image was extracted from [Pena et al. 2020]**

A group of consumers (CG1) then, each, fetch the corresponding video file and split that into several video scenes, using the PySceneDetect [Castellano 2012], that are persisted back into another bucket (”staging”) on Cloud Storage.

From that point on, another Cloud Function is triggered when new files are pushed into the staging bucket, which leads to these video scenes metadata being pushed to another Kafka topic (raw-video-scenes-refs).

Another consumer group (CG2A) fetches and loads the raw video scenes, to perform the face detection and recognition to generate the embeddings. That architecture enables yet another consumer groups to fetch exactly the same video scenes to perform scene classification, or even 3D pose estimations.

The generated embeddings are finally persisted on yet another GCP Bucket, which then triggers another Cloud Function to push it into a 3rd Kafka Topic that persists it into a database.

Figure 3 illustrates the pipeline in which each step happens.

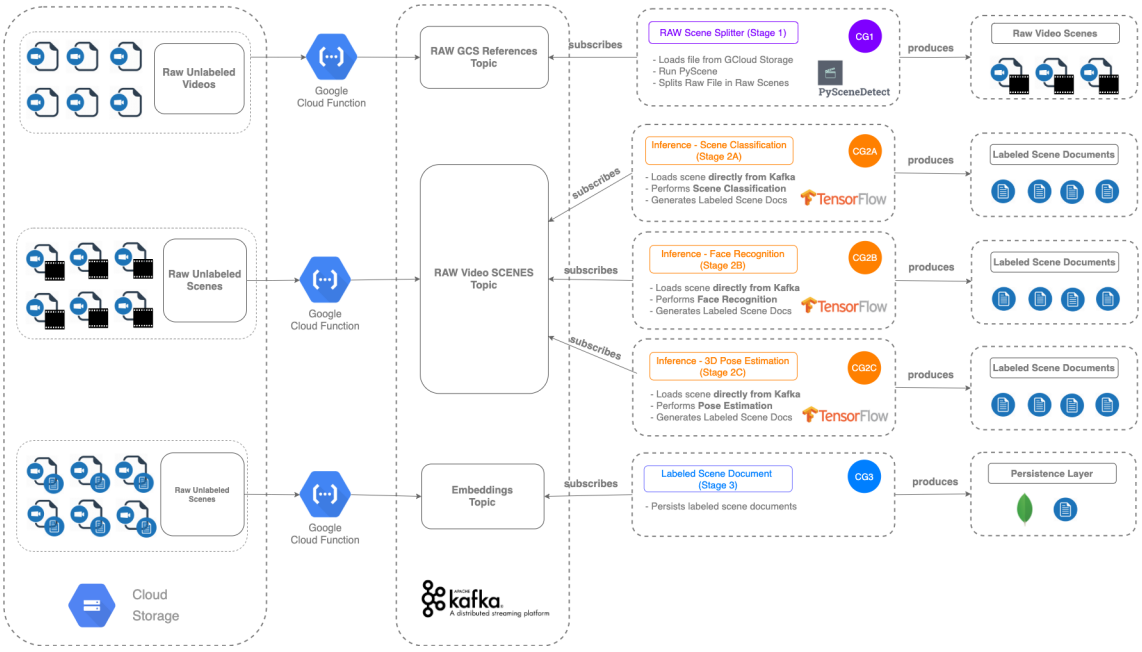


**Figure 3. Reglobinition (Globo Face Stream) Architecture Proposal**

## 2.4. Conclusion

When it comes to video metadata generation, the proposed method achieved important goals. A huge archive of soap opera videos will be enriched with metadata

enabling different application improvements. New features for recommender systems and user's preferences information to help advertising targeting are the first two company's applications that will benefit from this metadata. Currently, the facestream [Pena et al. 2020] solution pipeline is being adapted to work in this proposed architecture for massive video processing. In addition to this work, other solutions such as object detection and action localization in video scenes will also be evaluated in this architecture.

## References


Castellano, B. (2012). Pyscenedetect. *Journal Title*, 13(52):123–456.

Pena, R., Ferreira, F. A., Caroli, F., Silva, L. J. S., and Lopes, H. (2020). Globo face stream: A system for video meta-data generation in an entertainment industry setting. In *ICEIS*.

Sax, M. J. (2019). Apache kafka. In Sakr, S. and Zomaya, A. Y., editors, *Encyclopedia of Big Data Technologies*. Springer.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832.