Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in
Statistica per le Tecnologie e le Scienze

Relazione Finale

Instagram Images and Videos Popularity Prediction: a Deep Learning-Based Approach

Relatore: Prof. Gian Antonio Susto
Dipartimento di Ingegneria dell'Informazione

Correlatore: Luca Brunelli
Statwolf Data Science

Laureando: Massimiliano Viola
Matricola N. 1224874

Anno Accademico 2021/2022

*To my family, which has always supported me,
and to all my friends, relatives, and mentors of all ages,
that constantly push me to become a better person.*

Abstract

In recent years, social media platforms have seen tremendous growth in terms of the number of users, forms of interaction, and diversity of content. While these channels are purely a source of entertainment for many users, for others they represent the main source of revenue or advertising for their products and services. In order to capture users' attention, companies and professionals aim at achieving high popularity of their posts. In this work, we aspire to predict post popularity on the Instagram platform through Machine Learning approaches, with the goal of presenting a methodological tool that could provide useful information for post performance optimization. While previous contributions on the subject addressed the generic popularity of a post on the platform, we focus on the post popularity on a specific profile using only the visual content related to the post (image or video). We describe in detail the process and workflow to design a measure of popularity consistent even over the long time frame. Furthermore, we take advantage of state-of-the-art Convolutional Neural Networks and provide interpretability traits for their predictions, a quality that is nowadays highly welcomed in the industry. Lastly, we use a situation of scarce video data to experiment with ways of performing mixed training with both images and videos, providing problem-independent ideas and architectures that can potentially be applied to other video classification tasks.

Keywords: Computer Vision · Convolutional Neural Networks · Popularity Prediction · Social Network.

Preface

This work is the report of the internship experience carried out in collaboration with Statwolf Data Science, supervised by Prof. Gian Antonio Susto and Luca Brunelli respectively as university and company tutors. The objective of the internship, which took place from June to September 2021, was to study the feasibility and possibly proceed with the prototyping of a system capable of extracting relevant information from posts on the main social media platforms through Computer Vision methods. To achieve this goal, the following activities were carried out: (i) collection of metrics and images/videos relevant to the task thanks to the corporate BI platform; (ii) review of the main state-of-the-art architectures and Convolutional Neural Networks; (iii) design of new types of neural networks ad hoc for the task; (iv) feature engineering and interpretation of the results. The content of this thesis was presented by the authors on November 30, 2021, at the 1st Italian Workshop on Artificial Intelligence and Applications for Business and Industries (AIABI 2021), co-located with the 20th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2021). The proceedings of the workshop are available at `http://ceur-ws.org/Vol-3102`. Thanks to Prof. Susto, Luca Brunelli and the whole Statwolf team for the very formative experience in an innovative, collaborative and flexible work environment.

Contents

List of Figures

1 Introduction

Social media has become, in recent years, fundamental platforms for marketing and advertising; post popularity is considered as a good proxy of marketing strategy success in social media: for this reason, predicting post popularity is not only of interest from an academic point of view, but also crucial from a business and marketing perspective. With the possibility to reach thousands of users with ease, consistently posting the right trending content can translate into a significant increase in follower interaction and consequently sales for an emerging or even established brand.

In this work, we aim to predict Instagram post popularity via Machine Learning (ML) approaches, with the goal of presenting a methodological tool that could provide useful information for post performance optimization. In the proposed approach we exploit state-of-the-art *Convolutional Neural Networks* (CNNs) and provide interpretability methods for their predictions. Lastly, we use a situation of scarce video data to experiment with ways of performing mixed training with both images and videos, providing problem-independent ideas and architectures that could potentially be applied to other video classification tasks.

The rest of the work is organized as follows: in Section 2 we provide literature review on post popularity prediction and we propose a new metric, called *Popularity Rate*; moreover, in Section 3 we formalize the ML task at hand. In Section 4 the proposed approach is presented, while Section 5 is devoted to detail the experimental part of this work: the real-world dataset employed, the experimental settings and results. Finally, Section 6 reports the conclusions of this work and discusses potential future research directions.

2 Proposed Popularity Metric

On the Instagram platform, *likes* and *comments* are arguably the most quantifiable components of a post's success. Despite this, there is no universal measure of *popularity*, and the choice of the metric used to describe it, starting with the above ingredients, is itself an interesting subject for studies.

In recent works on the topic, [9] and [15] considered dividing the sum of likes and comments by the number of followers of the profile and treated the problem as a regression task, whereas [16] formalized it as a binary classification by taking the best and worst 25% of each user's posts sorted by number of likes. Finally, also [1] considered two popularity classes, but used a moving average window on the likes trend (defined as *Likes Moving Average*) to dynamically determine if a user's post performed better or not than its latest K predecessors.

All of the aforementioned works aimed to measure the popularity of a post in absolute terms within the Instagram platform, using multi-user datasets often in combination with contextual information. In our work, however, we have the goal of interpreting the popularity within a specific profile, wanting to be as accurate as possible in predicting how a well-defined audience would respond to a given post, allowing practitioners that decide to use our method to choose accurately the content to publish.

Focusing on the single Instagram profile implies that we need to look over a very long time frame in order to collect enough data, which intrinsically leads to major obstacles that did not exist in the metrics used in previous works. Indeed, in these settings absolute values are typically not meaningful and/or reliable: the follower count used to normalize across different profiles is not a reasonable quantity when aiming at modeling a single profile behaviour, while likes and comments grow by various orders of magnitude if the time interval is not restricted, making the most recent posts always the most popular. In fact, working over the long term, it certainly makes more sense to normalize for a metric which depends on time when the post was published, as suggested by [17].

From a business point of view, an important metric is certainly the *engagement*, obtained by dividing the sum of likes and comments of a post by its total views (also known as *impressions*). This metric is crucial to keep track of sponsored posts, which might occur frequently for a brand or business profile, as well as increasing the predictive potential by providing more and exact information. The importance of this data can be observed for example in [5], which uses the impressions directly as a metric given appropriate auxiliary information. Unlike other social media platforms, however, the number of impressions is not publicly available to download via the Instagram API.

Even with the privileges to access private metrics (as in the case of impressions), it's not easy to retrieve data from past years. In many cases, it is necessary to rely on third-party applications to do this, but generally, they can fetch the various metrics up to 2 years in the past. Since not all entities have been far-sighted in this context, typically data of the impressions is known for a very short period of time, thus, using impressions would force us often and willingly to limit a lot the time interval in which to collect data from a specific profile.

Returning therefore to the idea of discounting likes and comments not by the number of impressions but rather by the number of followers, we propose the following metric for popularity of a post p, referred to as *Popularity Rate* (PoR):

$$\mathrm{PoR}(p) = \frac{l(p) + c(p)}{f(t_p)} \tag{2.1}$$

where $l(p)$ and $c(p)$ are respectively the likes and comments of a post p, while $f(t_p)$ are the followers of the profile at the time t_p the post p was published.

Since the number of followers at upload date is not an attribute of the post, it does not come with the associated metadata, we recommend exploiting follower trends provided by external services that track Instagram metrics[1]; we resorted to interpolation of the known data points provided by the external service in order to have an estimation of $f(t_p)$. It may happen that even in this way, the number of followers in the past is not available until the desired date, but here, unlike the impressions, we can reconstruct the missing data in a robust way as explained in [15].

In Fig. 2.1.a, for the case study that we are going to describe in detail in Section 5, we compare the division of the sum of likes and comments by the number of followers and by the total impressions in a time period in which both metrics are available. Since we observe a good linear correlation (0.806) between them over a two-year period, this shows the substitute metric is trustworthy: we tolerate a small error in order to be able to extend the number of posts that we can consider and exploit in our ML approach.

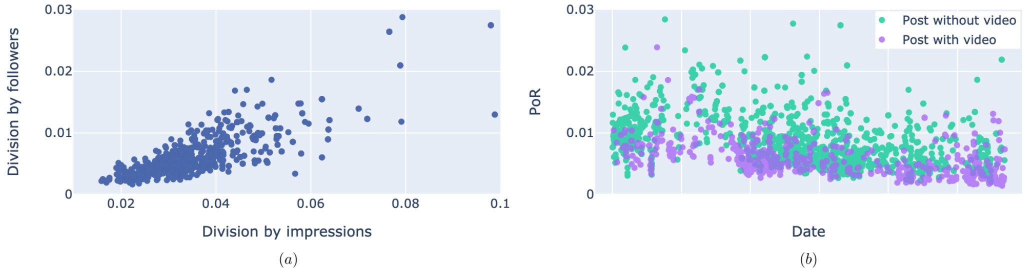


Figure 2.1: Panel (a): comparison between dividing the sum of $l(p)$ and $c(p)$ by the total impressions vs. the $f(t_p)$. Panel (b): effect of the presence of video content on the PoR.

[1] In this work, we have used the web service Not Just Analytics [8].

3 Problem Formalization

At this point, we make the choice to tackle the problem as a classification task, rather than a regression one, as it is typically done in the literature [1, 16]. This choice is motivated by two main factors: (i) typically for social content managers/creators it is sufficient to have classes of popularity associated with posts, without high level of granularity; (ii) on a ML perspective, the classification formalization makes the problem treatable, as precise regression models could be difficult to be developed in this context.

We exploit the PoR defined in the previous Section to derive *Popularity Classes* (PoCs), ie. non-overlapping classes that are defined on the PoR metric as a discretization of such continuous quantity. In this study, we will consider both the classification problem with 2 PoCs and with 3, but the same procedure can be applied with any problem cardinality.

Initially, we defined the PoCs by equally dividing all the posts: using the median and the terciles of the PoR distribution as splitting points to delineate, respectively, the 2 and 3 classes labels. After this operation, we realized that a lot of the top-class posts were very old, whereas the latest ones were not as popular; this was a direct consequence of the fact that the PoR is generally higher when a profile has fewer followers. This phenomenon is described in [9] and [15] by comparing the average PoR for different profiles: this downward trend is assumed to be due both (i) to the fact that early followers are the most interested in the content and (ii) to the growth of the platform itself which naturally exposes users to more content and gradually reduces interest in a specific profile.

For this reason, inspired by [1], we divide into the 2 or 3 PoCs using again median and terciles, but this time evaluating these statistical indicators within a rolling time window of several weeks. Such procedure is done under the assumption that a local label assignment is better than a full-time horizon one, since: (i) a post in a certain time frame is only compared with the nearest ones under very similar environmental conditions; (ii) in this way, we are more robust to errors and anomalies in calculating the follower estimate since the effect of a bad evaluation is only observed locally. The direct comparison of the two alternatives, on the experiment of Section 5 and for the 3 classes problem, is shown in Fig. 3.1.

Everything explained so far would be done separately for images and videos: we make this choice because, after evaluating several Instagram profiles, we have seen that these two media have different behaviours in terms of PoR (and consequently PoCs). This is a key difference particularly evident in our case study, when we notice posts containing only image content perform on average

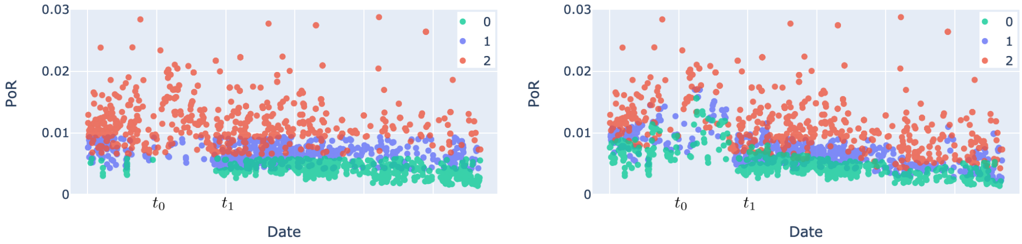


Figure 3.1: Comparison of label assignment in our case study (see Section 5) using a horizontal division (left panel) or a sliding time window of several weeks (right panel). On the right, it's clearly observable how a local anomaly (in the period t_0 to t_1), due to our erroneous follower estimate or to a change in social media management policies by the profile administrator, is resolved. Class overlap in the figure is due to image and video data points displayed together.

45% better in PoR than posts with at least one video (see Fig. 2.1.b that refers to the case study of Section 5).

Since we use the sliding window independently for videos and images, the result is a class-balanced dataset for both media type; this unfortunately has a drawback, since a mixed post with both images and videos could end up with different labels for different types of media.

4 Proposed Approach

4.1 Data Preprocessing

Once collected all the media related to each post (image or video with relative metadata) and created the PoCs, a cleaning operation of the dataset needs to be performed. The major problem is the presence of duplicate or near duplicate content, published at various times, that could lead to two different issues: (i) free predictions in the validation sets; (ii) conflicting labels in the training phase. Removal of duplicates is performed as follows.

For image content, we input pictures into a pre-trained CNN [4] and take the activations of the last layer before the classification head. Nearest Neighbors search is then performed in the embedding space of the CNN: every time two images have an Euclidean distance less than a certain threshold, they are declared duplicates and one of them is removed. A good initial guess for the threshold can be easily identified by plotting the histogram of the distances between one image and its closest neighbor, for each image, and picking the value that cuts the left tail off. By visually inspecting the duplicate pairs, this value can then be adjusted properly as needed (see Fig. 4.1.a as example for the case study of Section 5). For the video content, the same can be done cheaply but effectively by looking at the video thumbnail and attributes such as length and frame rate.

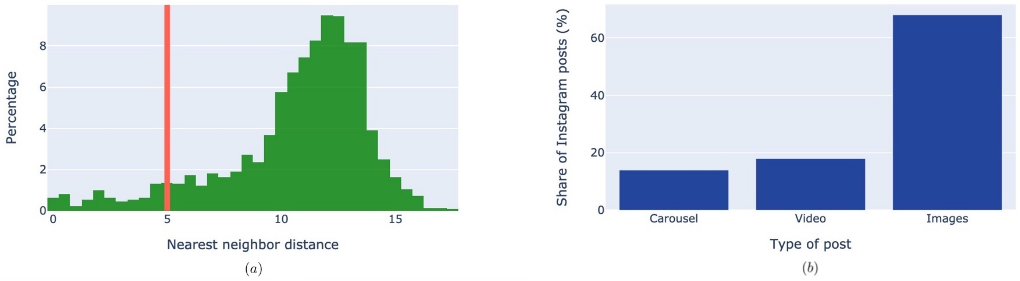
Whenever two duplicates are found and their labels are different, priority to single posts over multiple ones (carousels) is given. In case both are of the same type, the most recent one is always preferred.

Carousels are not very common as a type of post (Fig. 4.1.b), but since they include various media simultaneously they can be important to enrich the training set, so we suggest inspecting the data and leave only the relevant content.

4.2 Classification

After performing all the preprocessing steps explained above, we obtain the final dataset, on which we can now train various classification models to predict the PoCs of a post. In the following, we present a series of architectures that allow us to handle images and videos with a separate or a mixed approach.

We first show a solution that uses only posts with images, being in general the most frequently occurring type, as can be seen in Fig. 4.1.b. Although videos



(*a*)

(*b*)

Figure 4.1: Panel (*a*): histogram of the closest distances from one image to all the others, for each image, with duplicate image threshold cutting off the left tail of the distribution. Panel (*b*): distribution of posts on Instagram as of June 2019 [12], by type.

occupy a secondary role on Instagram, recently more and more companies have started to create content of this type. We therefore consider essential in this work to take them into account, because: (i) on a single profile, even over a very long period of time, the images published may not be enough to allow optimal training, so adding the (even few) videos allows us to increase the total size of the dataset and may be useful in a modeling perspective; (ii) it's definitely a plus for companies to have a metric for evaluating video content as well.

Thus, we then show a model based only on videos, to be used, given the scarcity of data, as a benchmark, and finally we implement two different *mixed* solutions with the objective of improving results on videos by taking advantage of the image dataset which, as mentioned, generally has much larger size. The way to do this in a situation of scarce training data for one of the different source, however, is not a cutting-edge research field. The only work we were able to find on the topic dates back to 2015 [13] and explores the idea to do *Transfer Learning* from images to a video recognition task. We believe the reason for such little interest in recent years is due to the fact that modern pre-trained CNNs have become so robust that image classification or simple video recognition can be performed and solved on a small scale, sometimes even looking at a single frame of the sequence, without needing to combine the resources.

Moreover, our situation is very peculiar also because the type of videos we are dealing with are not smooth, meaning we find rapid transitions of scenes, light and dark effects, and a single frame may or may not tell something about the content of the video. Nevertheless, we propose 2 different approaches that differ in the methodology used to adapt one type of data to the other.

4.2.1 Image Classification

As said, working, even if on the images, on a single profile, does not allow us to have enough data to train a CNN from scratch, for this reason we use a pre-trained model and then we perform a Transfer Learning procedure from it. Regarding the choice of the pre-trained model, we opt for the EfficientNet [14] family for their performing speed and size. In particular, the smallest model EfficientNetB0 pre-trained on the ImageNet [2, 10] dataset is what we eventually use, since early trials showed that increasing the complexity seems not to bring any significant benefices in terms of accuracy. Dropout with a 0.2 rate is applied in-between the pre-trained backbone of the EfficientNetB0 and the classification layer with either one or three outputs, depending on the number of classes we use. Medium data augmentation is performed to further regularize: horizontal flips, random translations up to 10% in both height and width direction, random 10% zoom, random brightness, contrast, and saturation changes. Adam optimizer with 0.001 learning rate is used in combination with standard cross-entropy loss, while the image size is set to 224x224. The models are trained for 15 epochs keeping the base not trainable to avoid overfitting.

4.2.2 Video Classification

To classify the video content, we choose a popular hybrid architecture [7], with a pre-trained CNN that extracts meaningful spatial information from the video frames and a *Recurrent Neural Network* (RNN) that models the temporal relationship between them. Known as CNN-RNN, this method generally performs very well because it is simply based on the assumption that a video is nothing but an ordered sequence of frames (images). To achieve this, videos are preprocessed by taking one frame every second for the first 15 seconds: if the video is too short, the time interval between frames is reduced. Using an EfficientNetB0 to extract feature vectors from the resized 224x224 frames, the input size has shape 15x1280. The architecture briefly described earlier consists of a couple of *Gated Recurrent Unit* (GRU) [4] layers with 8 and 6 units respectively, the first one returning sequences, then a dense layer with 8 neurons and the classification head. Regularization is applied via 0.2 dropout inside the GRUs and before the dense layer of shape 8. Optimizer, loss function, number of epochs are the same as before and we won't repeat this detail from this point forward.

4.2.3 Image to Video

The first mixed approach is based on the idea that we can think of an image as the minimal representation of a static video with all equal frames. For this reason, it makes sense to take the exact same CNN-RNN architecture used with video content only and increase the number of training samples as a regularization factor by using static frame sequences generated from images. We are aware that this monotony is not well represented within real videos, but this is done mostly to improve spatial rather than temporal information.

4.2.4 Video to Image

The second approach is the opposite as before and it is a video-to-image one, meaning we try to summarize the information within a video by working around the time component and focusing majorly on the spatial one. The idea we propose is about creating video embeddings that have the same shape as those from one image, and then train a model using these features on the combination of both. In order to do this, we first load and preprocess the video frames as we have done so far for CNN-RNN-based architectures, then we reduce the temporal relation dimension by applying an aggregate function to the time axis, leaving us with an embedding vector of shape 1280 for each video. Regarding the last step, we find taking the maximum to be the most effective among the standard aggregate functions, in the same way that a maximum pooling is often preferred to an average one in CNNs. In terms of convolutions, doing this operation means taking the maximum value of a certain pattern or feature map in frames during the whole video, claiming extreme values are the ones that give a reasonable representation. Of course, the more the video is static, the more this feature vector resembles a single frame, while if the video involves a lot of different scenes, this becomes more difficult to interpret. To classify, we opted for simplicity to use a single dense layer with one or three outputs with a 0.2 dropout, just as we did for images: our prediction is thus the activation function of the weighted sum of these features.

5 Experimental Results: a Real World Case Study

We tested the proposed approach in a real-world scenario, with the contribution of a leading trademark that works in the production of sports and leisure equipment who gave us access to its Instagram profile, which has long been active on the platform. Using the metadata collected from the profile, and the follower trend obtained from an external service, we were able to build a 5-year long dataset that we used to calculate, using (2.1), the PoR, which was then divided before into 2 and then 3 PoCs. After that, given the 1613 images and 575 videos related to the various posts, we performed the preprocessing by: (i) loading and resizing images and thumbnails to 224x224, extracting 1280-dimensional feature vectors from an EfficientNetB0 and finally opting for a threshold of 5 (see Fig. 4.1.a); (ii) manually inspecting carousels to exclude logos or product descriptions. Preprocessing eventually discarded 144 images and 19 videos, leaving us with a pretty class-balanced dataset of 1469 pictures and 556 videos, for a total of 1449 unique posts.

5.1 Preliminary Analysis

With comparable labels, a newsworthy analysis we could do was look for patterns in the post history. We wondered if similar images had similar labels and if there were types of images and products that people particularly liked or disliked. For answering these questions, we took images and video thumbnails, extracted the embeddings from an EfficientNetB0 in the same way we explained in 4.1, and then projected the high-dimensional vector of features into a 2d space using t-SNE [6]. In Fig. 5.1 we present the results for three classes: impressive was the ability of the pre-trained EfficientNet to identify characteristic product shapes and implicitly group them.

Searching for parts of the plane where the mean of the labels of a significant number of points was very high or low, we found some regions that confirmed the presence of popular or unpopular patterns in the image content. At the same time, the noise is very visible and the two extreme classes are often adjacent.

5.2 Validation Scheme

With such small image and video datasets like ours, validation became challenging due to the non-negligible variance of the results even for a fixed set

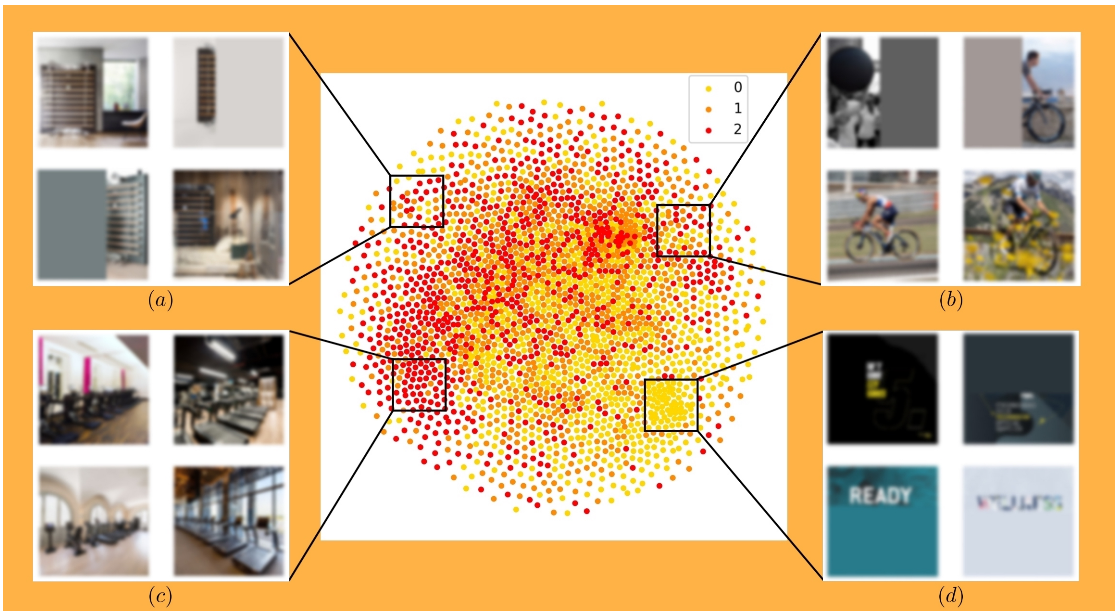


Figure 5.1: t-SNE projection of images and video thumbnails from the dataset in question, blurred and obscured in some sections for confidentiality. (a), (b), (c), and (d) show, each at a different point in space, examples of nearby images. These regions are clearly similar in terms of visual features: in (a) we find a horizontal line pattern, in (b) round patterns, in (c) a series of the same object, and finally in (d) various words.

of parameters and seed. For this reason, a single *StratifiedKFold* with 5 splits was generally not stable, and so we chose to run each experiment three times as our validation scheme. The metric we monitored in each run was the total accuracy over the 5 splits and we averaged the three results as a final measure of performance. During training, for each fold and in each of the three runs, a callback saved the weights of the model with the best validation accuracy. While this setup still left room for uncertainty, we believe it reduced it enough to safely compare the results across different experiments and model architectures.

5.3 Image Classification

Validating in the way described earlier, we achieved an average accuracy of 0.54 with three classes and of 0.72 for the binary classification task, with a top-2 accuracy for the multi-class model peaking at 85%. Considering that boundary labels both in the binary and multi-class cases were comprehensibly often confused by the nature of the task, the results seemed very satisfying. The confusion matrices of the image classification models are reported in Fig. 5.2: the standard deviations are computed between different experiments, not single folds.

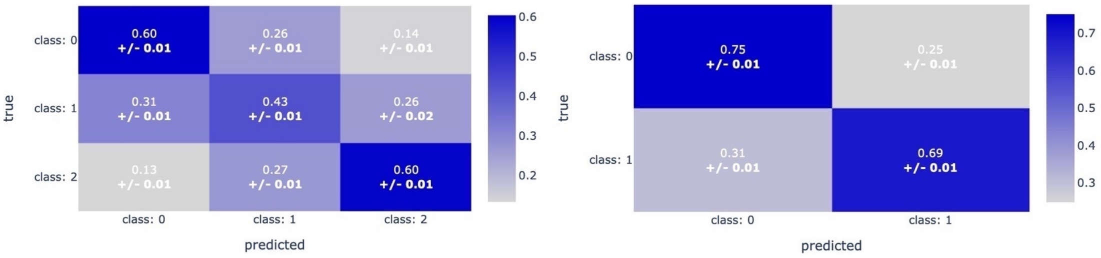


Figure 5.2: Confusion matrices: image classification.

5.4 Video Classification

Our case study greatly employed videos compared to many other profiles, so much so that in a particular 1-year period of time the fraction of video content reached a remarkable 43%. The total number of videos was 556, and in 30% of the cases we had to take more than one frame every second due to the videos being shorter than 15 seconds. We achieved an accuracy of 0.53 with the three classes and 0.70 for the binary classification task. The confusion matrices, reported in Fig. 5.3, make immediately clear that the multi-class model was quite poor and suffered from a low recall of the intermediate class, predicting very often only the extreme ones. Furthermore, this architecture had the substantial problem of being over-parameterized: the first GRU layer has 30960 parameters, which in combination with the dataset being small and having no way of applying data augmentation made training a trustable model really hard.

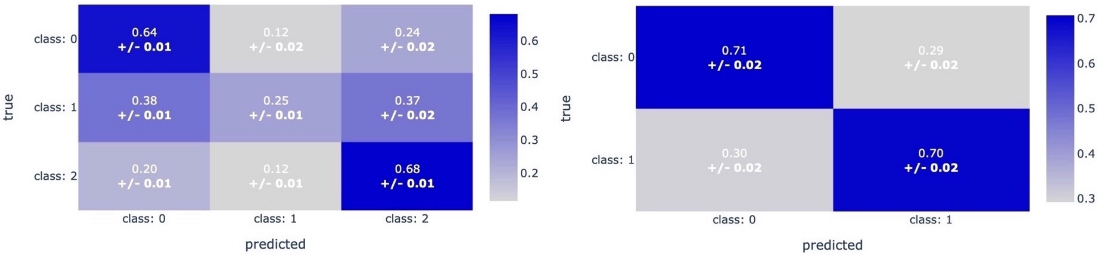


Figure 5.3: Confusion matrices: video classification with CNN-RNN.

5.5 Image to Video

Validating on the same folds of the video baseline, we achieved an accuracy of 0.54 for the multi-class problem and 0.70 for the binary classification task. While we did not observe the desired increase in accuracy, we can see from the confusion matrices in Fig. 5.4 that the predictions changed significantly. In particular, it seems like static videos generated from images made the normal video predictions shift towards the lower classes, reducing certain types of errors but introducing new ones. The takeaway of this experiment was that this kind of mixed training had a clearly visible effect on the predictions, and video classification could potentially benefit even from static sequences of frames generated from images.

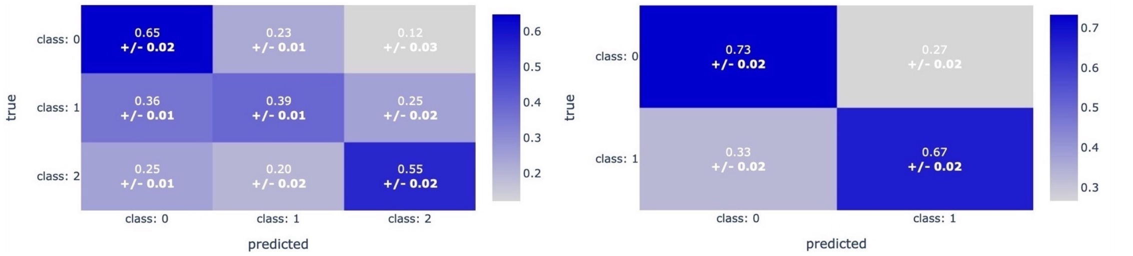


Figure 5.4: Confusion matrices: video classification with CNN-RNN including, during the training phase, static videos generated from images.

5.6 Video to image

Validating once more on the same video folds as the CNN-RNN, we achieved an accuracy of 0.54 for the multi-class problem and 0.71 for the binary classification task, with the corresponding confusion matrices reported in Fig. 5.5. Again, we did not see a significant improvement, but we were positively surprised by the results in this setup, considering that we were drastically reducing the number of parameters needed for video classification, as well as condensing temporal information, while maintaining the same performances.

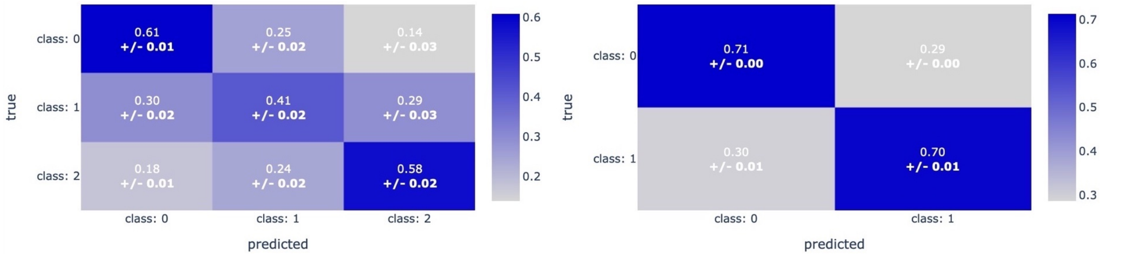


Figure 5.5: Confusion matrices: video classification using, during the training phase, both video and image embeddings.

5.7 Model Interpretability

Given the strong applicative and business-oriented nature of our work and research, we believed explaining predictions was as important as solid modeling.

In this section, we present some interpretability examples for image classification models obtained with Grad-CAM [11], a technique for producing visual explanations for decisions from CNNs. In a nutshell, the Grad-CAM algorithm creates a class activation heatmap to superimpose on the original image, which represents a coarse localization map highlighting the important regions for predicting the specific class. In Fig. 5.6, we can see it in action on three images [3]: these are not sampled from our dataset for confidentiality reasons but are similar enough to allow the comparison. The represented heatmaps refer to the top class when using a multi-class classification model: as we can see, the focus is well located on the meaningful components and areas.

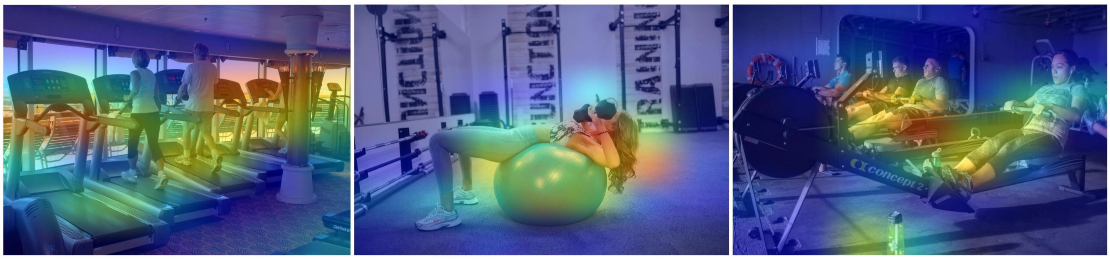


Figure 5.6: Model interpretability examples with Grad-CAM. Images from [3].

6 Conclusions and Future Works

In this work, we presented an approach to first define a popularity metric for an Instagram post using data always easily accessible and then to classify the popularity using Deep Learning-based models. We then showed various solutions that allowed us to also take into account the little but significant data derived from posts containing videos. Always keeping in mind that popularity classes are very noisy and thus not close to be perfectly separable, the results are very promising, in particular we demonstrated that we were able to significantly isolate the classes with low and high popularity.

Some future research directions are foreseen: (i) include historical/contextual information as a feature for various models: a particular image/video that was successful in the past is not necessarily successful at the present time (and vice versa); (ii) strengthen the training of video networks by applying data augmentation: instead of extracting the frame features before training, they can be processed in real time, augmented as if they were images; (iii) while the presented approach was designed for Instagram, we think that both the proposed metric and modeling pipeline can be easily extended to other social media platforms.

Bibliography

[1] Salvatore M. Carta et al. “Popularity Prediction of Instagram Posts”. In: *Inf.* 11 (2020), p. 453 (cit. on pp. 2, 4).

[2] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255 (cit. on p. 8).

[3] *From left to right: "Ocean View GYM" by Prayitno; "Young Woman exercising with a fit ball in modern gym" by shixart1985; "Sailors exercise in the gym aboard USS Dwight D. Eisenhower" by Official U.S. Navy Imagery. Licensed under CC BY 2.0.* https://search.creativecommons.org/. Accessed: 20/09/2021 (cit. on p. 14).

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 (cit. on pp. 6, 8).

[5] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. “What makes an image popular?” In: *Proceedings of the 23rd international conference on World wide web* (2014) (cit. on p. 2).

[6] Laurens van der Maaten and Geoffrey Hinton. “Viualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605 (cit. on p. 10).

[7] Joe Yue-Hei Ng et al. “Beyond short snippets: Deep networks for video classification”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4694–4702 (cit. on p. 8).

[8] *Not Just Analytics*. https://www.notjustanalytics.com/. Accessed: 20/09/2021 (cit. on p. 3).

[9] Kristo Radion Purba, David Asirvatham, and Raja Kumar Murugesan. “Instagram post popularity trend analysis and prediction using hashtag, image assessment, and user history features”. In: *Int. Arab J. Inf. Technol.* 18 (2021), pp. 85–94 (cit. on pp. 2, 4).

[10] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115 (2015), pp. 211–252 (cit. on p. 8).

[11] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 618–626 (cit. on p. 14).

[12] *Statista Research Department*. Accessed: 20/09/2021. URL: https://www.statista.com/statistics/605107/video-and-image-brand-posts-on-instagram/ (cit. on p. 7).

[13] Yu-Chuan Su et al. "Transfer Learning for Video Recognition with Scarce Training Data". In: *ArXiv* abs/1409.4127 (2014) (cit. on p. 7).

[14] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *ArXiv* abs/1905.11946 (2019) (cit. on p. 7).

[15] Katja Thömmes and Ronald Hübner. "Why people press "like": A new measure for aesthetic appeal derived from Instagram data." In: *Psychology of Aesthetics, Creativity, and the Arts* (2020) (cit. on pp. 2–4).

[16] Zhongping Zhang et al. "How to Become Instagram Famous: Post Popularity Prediction with Dual-Attention". In: *2018 IEEE International Conference on Big Data (Big Data)* (2018), pp. 2383–2392 (cit. on pp. 2, 4).

[17] Alireza Zohourian, Hedieh Sajedi, and Arefeh Yavary. "Popularity prediction of images and videos on Instagram". In: *2018 4th International Conference on Web Research (ICWR)* (2018), pp. 111–117 (cit. on p. 2).