VIDEO INTEGRITY TESTING USING MINIMAL LEARNING

Abhinau Kumar Venkataramanan

Department of Electrical and Computer Engineering, UT Austin

ABSTRACT

Content Integrity is an important problem for social media platforms, since they rely on user-generated content. Users may attempt to game the platform's recommendation algorithms by presenting their content disingenuously by mimicking other types of content which the platform considers ”better”. For example, images may be presented as videos, which we call ”static” videos, to increase their reach on a platform like Facebook. This work proposes a method based on transfer learning to classify videos as static or non-static (authentic) using minimal learning, and shows its effectiveness on user-reported static videos.

Index Terms— Video Integrity, Static Videos, Transfer Learning

1. INTRODUCTION

Social media platforms intentionally prefer some forms of content over others. For example, Facebook prefers videos over photographs and text-posts, while YouTube prefers content creators who upload videos more often. This preference often takes the form of higher rankings in the recommendation algorithms used by the platform. As a result, content creators are encouraged to make certain types of content.

However, not all creators respond as expected. Some just want quick views and likes without dramatically changing the nature of the content they produce. So, they try to game the algorithm into recommending their content by spoofing the desired content. Specifically, in the case of Facebook, this takes the form of creating videos which essentially just have one image repeated over time. Since exactly identical frames are easy to detect, small artefacts are often added so that the video is not exactly constant over time.

It is important to detect and remove such static videos for two main reasons. Firstly, since these videos are upvoted by Facebook's ranking algorithm, they flood users' feeds, and many users find such low-quality content annoying. Secondly, such videos take views away from authentic videos from creators who actually invest in making high-quality videos.

2. EXISTING SOLUTION AND SHORTCOMINGS

Facebook's solution to the problem was to train a deep neural network with Spatio-temporal convolutions called R(2+1)D [1] (developed in part by Facebook Research) specifically to identify such “static videos”. The model was trained on videos found on Facebook and achieved good accuracy.

However, as videos of a certain type began to get taken down, creators changed the type of artefacts they were introducing to their static videos. This meant that Facebook had to retrain their model repeatedly as the content evolved. This reflects are more general problem of neural networks trained end-to-end, which is generalization to unseen data.

Secondly, this model stands alone alongside potentially many other models used by Facebook to analyze other aspects of videos uploaded to its platform - like objects and actions. So, when deployed at scale, simply detecting whether a video is static incurs a significant computational load in addition to other necessary analysis.

3. PROPOSED METHOD

The aim of this project is to achieve the same goal (binary classification of videos into static and non-static videos) but with much less dedicated training, by using Transfer Learning [2]. As the nature of the static videos evolves, only the small classifier at the end needs to be retrained. This saves a great deal of computation over time.

Secondly, since pre-trained models are used for feature extraction, this allows for significant feature sharing with other models which are already being used for Video Analysis. This drastically decreases the additional computational load during inference.

Finally, there is good precedent for comparing images in the embedding spaces of pre-trained networks, such as for style transfer, super-resolution [3] [4] and stereo matching [5]. The distances between embeddings (or other feature-maps) are called perceptual losses, and perform much better than per-pixel losses like Mean Squared Error (MSE) when comparing image content. In the subsequent sections, the exact details of the models used and the experiments that were conducted have been described.

4. DATASET

Real world data was collected from Facebook and other social media platforms. 60 user-reported static videos were obtained from a Facebook group dedicated to reporting such videos, and 60 non-static videos were obtained from Facebook, YouTube and Twitter, mostly of marketing videos, how-to videos and vlogs. This is representative of normal non-static content on social media. This resulted in a total of 120 videos.

200 frames were extracted from each video, from frame 51 to 250. This corresponds, approximately, to a segment of the video from second 2 to second 10. This was done to avoid introductory information like title cards, while also extracting a representative section of the video.

5. FEATURE EXTRACTION

Each frame of the video was first projected onto an embedding space using a pre-trained Deep Learning Network (DLN). Experiments were conducted using nine DLNs trained on the ImageNet database, to show the universality of the proposed features. After passing the video (frame-by-frame) through the DLN, we have a sequence of vectors corresponding to a trajectory in the latent space. Two types of features were extracted from this trajectory - drift and spread

”Drift” features attempt to capture information about the change in content between successive frames. For static videos, we expect successive frames to be very close in the latent space, while non-static videos will have larger differences between successive frames.

”Spread” features attempt to capture the diversity in the content of the video using the spread of these points in the latent space. For a static video, we expect all frames to be clustered closer to each other in comparison to a non-static video. These features do not depend on the order in which the frames were observed. This allows us to account for static videos where the ”noise” added is periodic in nature. This may lead to higher differences between successive frames, but on the whole, the points will still be clustered together.

Finally, if the embeddings of the frames have high magnitude, their differences will also be higher, which might lead to unfair comparisons. So, normalized versions of distances were also used as features. The exact features that were computed are detailed in the equations below. Let $\{\mathbf{X}_i\}_{i=1...F}$ denote the set of embeddings for each frame in a video having F frames (Quantities expressed in bold are vectors). Then,

$$\boldsymbol{\mu} = \frac{1}{F}\sum_{i=1}^{F}\mathbf{X}_i \tag{1}$$

$$s_1 = \frac{1}{F}\sum_{i=1}^{F}\|\mathbf{X}_i - \boldsymbol{\mu}\| \tag{2}$$

$$s_2 = \max_{i=1...F}\|\mathbf{X}_i - \boldsymbol{\mu}\| \tag{3}$$

$$s_3 = \frac{1}{F^2}\sum_{i=1}^{F}\sum_{j=1}^{F}\|\mathbf{X}_i - \mathbf{X}_j\| \tag{4}$$

$$s_4 = \max_{i,j=1...F}\|\mathbf{X}_i - \mathbf{X}_j\| \tag{5}$$

$$d_1 = \frac{1}{F-1}\sum_{i=2}^{F}\|\mathbf{X}_i - \mathbf{X}_{i-1}\| \tag{6}$$

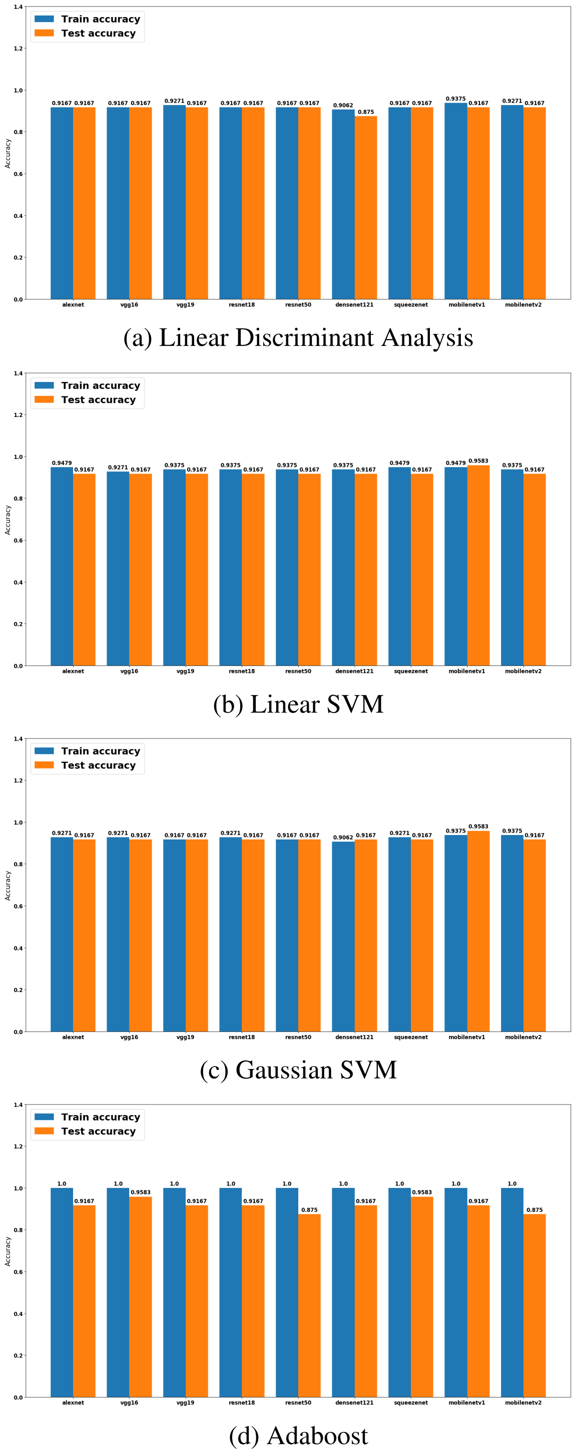
$$d_2 = \max_{i=2...F}\|\mathbf{X}_i - \mathbf{X}_{i-1}\| \tag{7}$$

$s_1 \ldots s_4$ are four spread features while d_1 and d_2 are the two drift features. 6 more features normalized features are obtained by dividing each of these by $\|\boldsymbol{\mu}\|$, giving a total of 12 features per video.

6. EXPERIMENTS

Nine DLNs were used to extract features - AlexNet [6], VGG16, VGG19 [7], ResNet18, ResNet50 [8], DenseNet121 [9], SqueezeNet [10], MobileNetv1, MobileNetv2 [11]. Four models were trained on features from each model - Linear Discriminant Analysis Classifier (LDA) [12], Support Vector Machine [13] with Linear Kernel (Linear SVM), Support Vector Machine with Gaussian Radial Basis Function Kernel (Gaussian SVM) and Adaboost [14].

Each classifier was tested on the six unnormalized features, six normalized features, and all twelve features. To avoid overfitting, model was trained using five-fold cross validation, with the median accuracy being reported. This was repeated 500 times for each model (since shuffling is random), and the median median accuracy is reported for each model.



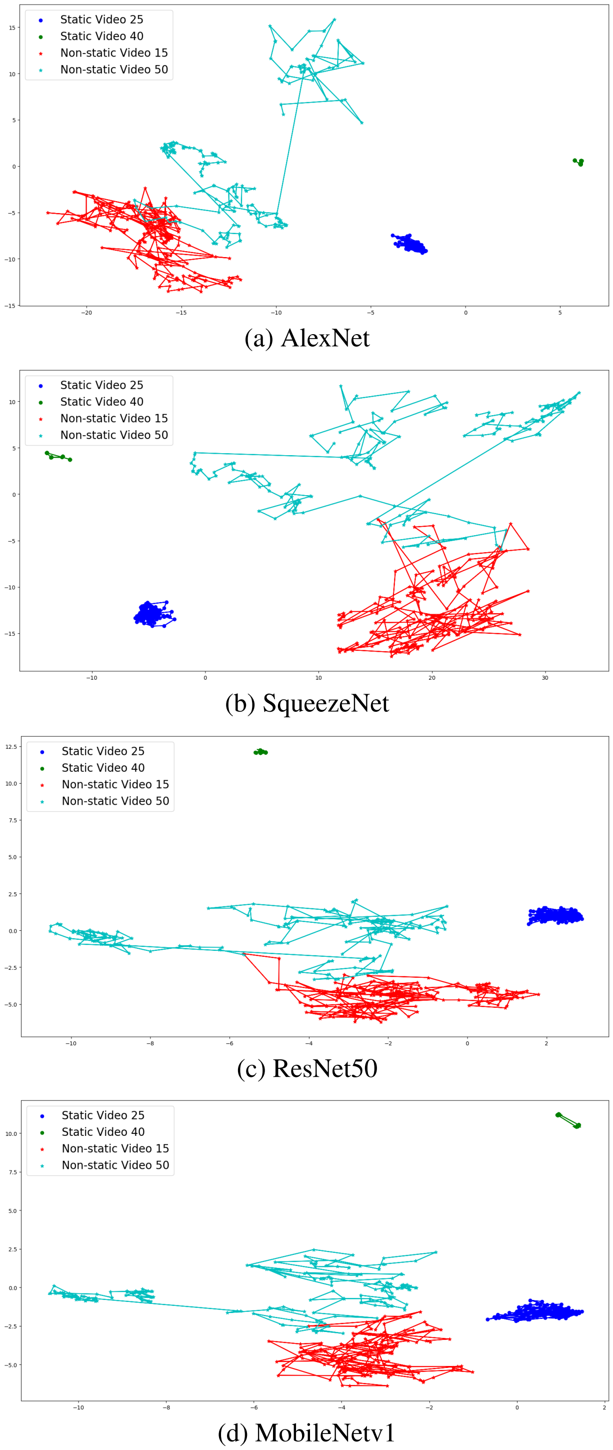
(a) Linear Discriminant Analysis

(b) Linear SVM

(c) Gaussian SVM

(d) Adaboost

Fig. 1. Performance of various classifiers by pre-trained model



(a) AlexNet

(b) SqueezeNet

(c) ResNet50

(d) MobileNetv1

Fig. 2. Visualization of video trajectories in 2D

7. RESULTS

The performance when using only normalized features was consistently and significantly worse than using unnormalized ones, suggesting that useful information is contained in the magnitude of the embeddings. They also provide little to no improvement when used in addition to the unnormalized features. The results of five-fold cross-validation when using unnormalized features are shown in Figure 1.

In order to visualize the validity of the features being extracted, 2D visualizations of the embeddings of two static and two non-static videos from four pre-trained models are shown in Figure 2. For embeddings from each pre-trained model, 2D representations were obtained by applying Principal Component Analysis (PCA) [15] on the entire dataset of embeddings and choosing the two principal components.

From these plots, we see that the performance of the different classifiers is nearly constant across pre-trained models used for feature extraction. In addition to performance, the trajectories for static and non-static videos exhibit similar properties across pre-trained models. This demonstrates the universality of the proposed method and suggests that it can be plugged into any existing deep video processing pipeline and use its features for inference.

8. CONCLUSION

In this work, we have proposed a method to distinguish between static and non-static videos using transfer-learning, and shown its effectiveness and its universality in choice of pre-trained models. So, the proposed algorithm has great scope for feature reuse when used as a part of a video analysis pipeline, leading to high performance with negligible additional computational cost.

9. REFERENCES

[1] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri, "A closer look at spatiotemporal convolutions for action recognition," 2017.

[2] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu, "A survey on deep transfer learning," *CoRR*, vol. abs/1808.01974, 2018.

[3] Justin Johnson, Alexandre Alahi, and Fei-Fei Li, "Perceptual losses for real-time style transfer and super-resolution," *CoRR*, vol. abs/1603.08155, 2016.

[4] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016.

[5] Jure Zbontar and Yann LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *CoRR*, vol. abs/1510.05970, 2015.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016.

[10] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[12] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien, "Linear discriminant analysis: A detailed tutorial," *Ai Communications*, vol. 30, pp. 169–190,, 05 2017.

[13] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, New York, NY, USA, 1992, COLT '92, pp. 144–152, ACM.

[14] Yoav Freund and Robert E Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[15] Jonathon Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.