Cinematographic Shot Classification through Deep Learning *

Bartolomeo Vacchetti
Department of Control and Computer Engineering
Politecnico di Torino, Italy
bartolomeo.vacchetti@polito.it

Tania Cerquitelli
Department of Control and Computer Engineering
Politecnico di Torino, Italy
tania.cerquitelli@polito.it

Riccardo Antonino
Department of Control and Computer Engineering
Politecnico di Torino, Italy
riccardo.antonino@polito.it

***Abstract*—Cinematographic shot classification assigns a category to each shot on the basis of the field size, which is determined by the portion of the subject and of the environment shown in the field of view of the camera. This task is very important in the context of the creative field and can help freelancers in their daily activities when it is performed automatically. Novel and effective approaches capable of processing large volumes of images/videos and analyzing them effectively are becoming increasingly important. This paper presents a data-driven methodology to automatically classify cinematographic shots through deep learning techniques.**

In our study, we consider four classes of film shots: full figure, half figure, half torso and close up and we discuss three different scenarios in which the proposed work can be helpful. A new dataset of images was created to evaluate performances of the proposed methodology and to compare them with state-of-the-art techniques. Experimental results demonstrate the effectiveness of the proposed approach in performing the classification task with good accuracy.

***Index Terms*—Machine learning, image classification, creative field.**

I. Introduction

The research activities discussed in this paper have been carried out to solve a problem of practical nature in the context of the creative field. People, who work in the creative field, usually deal with a considerable amount of unstructured data, such as images or video files, that are often unorganized and not classified properly. For example, in the video editing process, the editor has all the video files inside the same folder with no information regarding what type of content there is inside. Now the editor of the video has two choices. The first choice consists of using the video files even if they are disorganized, which implies that when the editor needs a new video a certain amount of time will be lost while looking for the file. This choice is somewhat efficient if the video files are few, however with more video files, the time that the editor has to waste looking for a specific file grows sharply. The second choice is to organize the video files manually, which is an operation that requires a certain amount of time too. Thus, both choices involve a loss of time that grows with the number of files considered. It is important to notice that with medium and big video production it is not thinkable to work with unorganized material, so if someone wants to have organized material it has to be sorted manually. The video editing is just an example to show the problem of unorganized material, which is an issue that emerges also in other areas of the creative field, from the arrangement of stock material to the creation of websites. In this scenario the need for effective and efficient data-driven engines capable of processing large volumes of unstructured data and analysing them effectively is becoming increasingly important. To this aim, machine learning algorithms can be used since they have the ability to acquire knowledge from a large amount of data and perform different kinds of data classifications/predictions based on the acquired knowledge.

This paper proposes a data-driven methodology to automatically classify cinematographic shots through machine learning techniques Among the different algorithms of supervised learning convolutional neural networks (CNN) are able to capture key properties in analyzing unstructured data such as images [1], [2] or videos [3]. Specifically, CNNs have had great success in large-scale image and video recognition, achieving state-of-the-art accuracy on classification and localisation tasks, also thanks to very deep architectures [1]. The main limitation of CNNs exploitation in many practical use cases is due to the large amount of data to train accurate models.

To overcome the above limitation, we decided to fine tune [4] a VGG-16[1] [1] to address the cinematographic shots, due to its excellent trade-off between training time and accuracy. The VGG-16 was pre-trained on the ImageNet dataset [5], which contains images belonging to 1 000 classes,ranging from vehicles to animals and people, and then fine tuned with our dataset, which contains the four classes of cinematographic shots considered for this study. We evaluated the proposed approach by running different experiments, changing the properties of the dataset under analysis and comparing the obtained performance with other models proposed in the literature. Specifically, we used two datasets with the same set of images, either monochrome or with an RGB profile. Such choice was made in order to investigate if and how

[1]The name VGG stands for Visual Geometry Group, the Oxford team that created this architecture, while 16 indicates the number of layer from which the network is composed.

978-1-7281-7303-0/20/$31.00 ©2020 IEEE
DOI 10.1109/COMPSAC48688.2020.0-222

Fig. 1. The 4 types of cinematographic shots considered for this study.

colours impacted the performance of the fine-tuned VGG-16 (see Section III-A1). We compared the performance of the proposed approach with others state-of-the-art methodologies to demonstrate the effectiveness of the fine-tuned VGG-16 in performing the classification task.

As a first attempt the network was trained only on images and not videos, but, since a video is nothing more than a sequence of images, if the network is able to classify images it might be generalized to work with videos. Someone may think that time, for instance, would be an extra feature to take into account, and generally it is, but not in this case. If a video shot is a *close up* it remains a *close up*, unless the video contains a camera movement, like the *dolly shot*[2]. In this scenario time becomes a relevant feature, but if the network is not capable of recognizing cinematographic shots it cannot recognize a camera movement.

This paper is organized as follows. Section 2 analyses in a deeper way the VGG-16 used, along with the presentation of some scenarios in which the proposed methodology for cinematographic shot classification could result useful. Section 3 discusses the preliminary experimental results, the dataset creation and the comparisons between the performance of the VGG-16 and the performances of other state-of-the-art methodologies. Section 4 discusses similar works available in literature, while Section 5 contains the conclusions and presents future research directions.

II. Tailoring the VGG-16 to the cinematographic shot classification

In this work we proposed to exploit a fine-tuned VGG-16 to address the cinematographic shot (i.e., images) classification on the basis of the field size, which is determined by the portion of the subject and of the environment shown in the field of view of the camera. The 4 types of shots considered for this study are shown in Figure 1. The discriminant in order to decide whether an image belongs to a class or another is the presence of the human figure or not and which portion of the human figure is shown. Unfortunately the cinematographic shot classification changes slightly from nation to nation[3]. We have chosen these four classes because they are taken into account by almost everybody. However, since the dataset used is small, we focused only on these four classes instead of considering other shots, such as the *american shot* or the *long shot*.Our classes are: the *full figure*, the *half figure*, the *half torso* and the *close up*.

The classification task has the role to learn from a given (historical) dataset a classification model to automatically identify the correct label for a new image. Since our dataset has small dimensions we have decided to use a pre-trained[4] model in and fine tune it. We have decided to do so because otherwise the small dimensions of the dataset could have led to overfitting. By fine tuning the model it is possible to exploit part of the ”knowledge” that it has acquired on another dataset and, after training part of the model on the new dataset, use that knowledge to solve a different task. In our case we have used a VGG-16 trained on the ImageNet dataset. The VGG-16 is a particular type of convolutional neural network (CNN). A CNN is a deep, feed-forward artificial neural network composed of many specialized hidden layers i.e. convolutional layers, pooling layers, fully connected layers and normalization layers. The concatenation of these types of layers multiple times lead to the creation of a deep convolutional neural network (DCNN), which has a state-of-the-art performance [1], [6], [7], such as the VGG-16.

A. Fine-tuned model building

The first step to fine tune the model is to remove the fully connected layers[5] on top and to replace them with new fully connected layers. At this point there will be all the remaining convolutional and pooling layers that still hold the knowledge gathered from the previous dataset. Before training the model on the new dataset some of these layers have to be frozen, in order to keep the knowledge gathered previously, while some others will be left ”unfrozen”. The ”unfrozen” layers will change their filters during the training phase on the new dataset, while the ”frozen” layers won't. By doing this the model keeps the ability to recognize simple patterns inside the images while at the same time it learns to recognize more complex patterns specific to the new images. Also the fully connected layers update their weights during the training phase with the new dataset. It was necessary to replace the old fully connected layers with the new ones because the VGG-16 trained on the ImageNet does not perform the cinematographic shot classification. After choosing how many layers to ”freeze” and substituting the fully connected layer the training phase

[2]It is a shot in which the camera moves either away or toward the subject filmed.

[3]It is sufficient to change the language on the Wikipedia page concerning the cinematographic shots and the number of classes changes.

[4]A pre-trained model is a model that is already trained to solve a specific task.

[5]The fully connected layers are the one responsible for the classification.

on the new dataset can begin. During the training phase only the "unfrozen" convolutional layers and the fully connected layers are affected by the back propagation algorithm and change, while the "frozen" layers remain unchanged. After the training phase the now fine tuned VGG-16 is capable of classifying images into cinematographic shots.

There are three main factors that justify the use of the convolution operator instead of a plain machine learning system, which are: (i) sparse interaction, (ii) parameter sharing, and (iii) equivariant representation. Due to the reduced dimension of the kernel, compared to the input shape, the CNNs interaction between the input and the output unit is sparse, which is very different with respect to traditional neural network layers.

B. Classification phase

The model resulting from the training phase is ready to classify new images into the different classes of cinematographic shots. When the fine tuned VGG-16 receives new images, the filters inside the convolutional layers slide across the new image creating feature maps that go to the following layer, that can be another convolutional layer or a pooling layer. The pooling layer extracts the most active values and gives them to the following convolution layer and so on until the flatten layer is reached. The non linear activation operation happens before the pooling operation and it is done in order to prevent an exponential growth of the values, which could potentially lead to evaluation errors from the VGG-16. The flatten layer reduces the dimensionality of the data to a vector that goes to the fully connected layers. Inside the fully connected layers, thanks to the weights, the data gets classified and is given to the output layer, the final layer, the one that makes predictions (more information concerning the specifics of the fine tuned VGG-16 created can be found at the beginning of Section III).

C. Application 1

The first scenario involves the reorganization of the material contained inside photographic archives. Suppose that a new set of old photos has been digitized. These new photos are intended to be used as stock material. Instead of labelling them one by one they could be labelled automatically.

D. Application 2

Let's now consider the scenario mentioned in the introduction with a little more background. After a full day of shooting all the video files are sent to the editor, who now has to edit the whole scene. He has his storyboard[6] to follow and now he just has to find the right video files to use. Unfortunately for the editor, the order in which the shootings were made does not coincide with the chronological order of the scene. Such a thing happens because on set there are a lot of people with different roles and commitments, so the shooting order is decided based on staff availability. Usually scenes in which more roles are needed take precedence, so when there is no further need for a professional role, for instance an actor, he or she is free to go[7]. Now if the scene is a short one the editor will have to deal with a few hundreds of video files, but if the scene is a long one, he or she will have to deal with thousands of video files. If the video files are already divided into their classes the time needed to find a video file is sharply reduced. This is just an example of a possible use of the cinematographic shot classification through deep learning, however if this type of classification is implemented with other types of classifications, such as interior exterior classification, or good shot bad shot it would be possible to reduce even more the editing time of a video.

[6]A storyboard is a sequence of images or illustrations that allows to previsualize a motion picture, an animation and so on...

E. Application 3

Another possible implementation of the cinematographic shot classification through deep learning is the genre recognition of a movie. Since the shots used in a movie have a narrative function, different genres of movies use a certain type of shot more frequently than another. For instance in a horror movie the *close ups* are usually a lot more frequent than the *long shots* due to the fact that the director wants to focus the attention of the viewer on the distressed expressions of the characters. However the ability of recognizing the frequency of the different types of shot by itself may not be sufficient in order to determine if a movie is a comedy, a documentary or something else. On the other hand if such information were to be used with other types of features, such as the color grading[8] and/or the soundtrack used, it would be possible for an algorithm to understand with certainty the genre of a movie.

III. Experimental results

Here we discuss the experimental evaluation of the proposed approach by exploiting two datasets including the same set of images, either monochrome or with an RGB profile. Dataset details are described in Section III-A. We first evaluate the performance of the proposed approach in terms of accuracy (see Section III-B) then, we compare our results with respect to the ones obtained by other approaches in the state-of-the-art (see Sections III-C).

The traditional architecture of a VGG-16 consists of 5 blocks made of two or three convolutional layers followed by a pooling layer. After these five blocks of convolutional and pooling layers there are three fully connected layers in charge of the classification, of which the last one is the one that implements the softmax activation and is the one in charge of making predictions. As previously stated, we have used a fine tuned VGG-16, so the last convolutional block was retrained with our dataset as well as the fully connected layers that follow. To evaluate the accuracy of the proposed approach we used the Stratified K-Fold Cross validation strategy (with K=10) and we analyzed the values

[7]This is also an advantage for the production because in this way it doesn't have to pay someone for a full day if he or she is needed for just a couple of hours.

[8]The color grading is a process that aims to strengthen the emotional impact of a scene through the use of colors.

of different metrics: *accuracy*, *precision*, *recall* and *f1-score*. While the accuracy evaluates the performance of the model globally, the others (very useful in presence of unbalanced datasets) are computed for each class separately. Specifically, precision is a measure of exactness since it represents the percentage of images labeled as belonging to class *c* that actually belong to it [8]. Recall, instead, is a measure of completeness because it captures the percentage of images of class *c* that are labeled as such. f1-score (also known F-measure), used to compactly summarize precision and recall metrics, is the harmonic mean of precision and recall.

The computer used to run the simulations was a MacBook Pro from 2018 with a 2.6 GHz Intel Core i7 6 core processor.

A. Dataset

The dataset used to train the VGG-16 was labelled manually for this study. The classes of film shot considered are the following: full figure (FF), half figure (HF), half torso (HT) and close up (CU). The number of classes of interest should be enriched easily by labelling a large number of images. As a first proof of concept of the proposed methodology, we labelled a set of cinematographic shots of 3 000 images. To balance the number of images with the number of classes we selected only 4 classes. In order to perform a classification with more classes a much larger dataset is required.

The labeled dataset of 3 000 images includes 750 *full figures*, 744 *half figures*, 758 *half torsos* and 750 *close ups*.

Half of the dataset was built through data augmentation by flipping horizontally the images. The original 1 500 images were gathered from different sources. More than half of the dataset, more precisely 840 images (56%), consists of movie frames downloaded from Internet, while the remaining images were photos. Out of those 660 photos, 425 (28.33%) were made by a professional photographer, while the remaining 235 (15.67%) were taken by an amateur photographer. The images gathered had different aspect ratios, however when they were fed to the network their shape had to be the same, so they were converted into images with an aspect ratio of 16:9. The 16:9 aspect ratio was chosen for two reasons. The first reason was that the 16:9 is the most used aspect ratio in both videos and images, so most of the samples had already such shape. The second reason, as shown by Kerns Powers[9], is that the other aspect ratios can be converted into the 16:9 without altering[10] the image too much. After reshaping the images, but before feeding them to the network, it was necessary to treat them a little more. The next step, after gathering, labelling and reshaping all the images, was to reduce their size, in order to reduce the number of features that the network had to take in as input. The size chosen was 160x90 pixels.

1) Monochrome vs Color Images: Since the colors inside an image are relevant but not determinant in order to understand if an image is *close up* or something else, we wanted to investigate how much the presence or absence of colors influenced the performance of the network. Thus, we used two datasets: (i) *DatasetRGB*, which had all the images with the original colors while (ii)*DatasetBW* includes monochrome (black and white) images. The monochromatic images use only the gray scale while rgb uses the red scale, the green scale and the blue scale. The values on the gray scale correspond to different shades of gray, with the 0 that corresponds to completely black and the 255 that corresponds to the absolute white. For instance, if all the values are zeros the image corresponding would be a completely black image. The fact that every value corresponds to a pixel is true only for monochrome images. With colored images things change a little bit. For instance, RGB images have three scales: the red scale, the green scale and the blue scale. So, when a computer reads an RGB image, for every pixel it receives three values, one for the red scale, one for the green scale and one for the blue scale. When the values of all the three scales are the same, the color resulting corresponds to a shade of gray, otherwise the color resulting depends on the magnitude of those values and the proportion between those values. We have used in both cases the VGG-16 to run the experiments. Since the VGG-16 needs to receive images with three channels as input we tripled the gray channel. By doing so the VGG-16 was able to work with monochromatic images.

With the *DatasetRGB* the VGG-16 accuracy grew up consistently with respect to the accuracy reached with the *DatasetBW*, which was 74.56%, reaching an overall accuracy of 81.29%. The increase in performance is mainly due to the detailed content of images in color with respect to monochromatic images.

B. VGG-16 performance

To evaluate the performance of the proposed approach we first run experiments on *DatasetBW* including four cinematographic shot classes: *full figure, half figure, half torso* and *close up*. Table I shows the classification report of the performed experiments by using a stratified K-Fold Cross validation strategy with K=10. The VGG-16 reaches an average accuracy of 81.30%, while precision and recall for each class are in the range 77%-90% and 76%-85%. Figures 2 and 3 show the trends of the accuracy and of the values of the loss function. While Table I reports the average performance of the 10 evaluated folds, Figures 2,3 and 4 refer to a fold that has an average performance w.r.t. the whole set. As the reader can see these are pretty good results considering the small dimension of the dataset used.

C. Comparisons with state-of-the-art approaches

Here we discuss the comparison of the proposed methodology with other types of state-of-the-art networks to better evaluate the quality of our fine-tuned VGG-16. We tested two state-of-the-art approaches: a multilayer perceptron (MLP) network and a generic CNN, with a simpler architecture compared to the VGG-16. Both ths MLP and the CNN were trained with the cross validation technique with a Kfold=50,

[9]The creator of the 16:9 aspect ratio.

[10]Here altering means distorting, removing or adding pixels to the image in order to obtain the 16:9 aspect ratio.

TABLE I
VGG-16 CLASSIFICATION REPORT

	precision	recall	f1-score	support
Full Figure	0.90	0.84	0.87	75
Half Figure	0.82	0.85	0.83	74
Half Torso	0.77	0.76	0.76	76
Close Up	0.81	0.81	0.81	75
Accuracy			0.81	300
Macro Avg	0.82	0.81	0.81	300
Weighted Avg	0.82	0.81	0.81	300

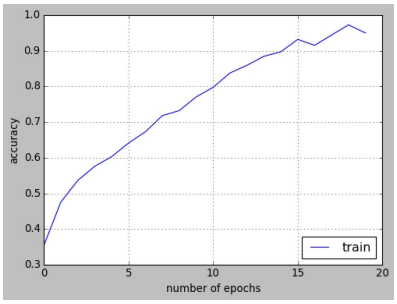

Fig. 2. Training accuracy

however every fold of the CNN was trained for 30 epochs, while the number of epochs for the MLP is 60.

The first comparison between the VGG-16 and the MLP concerns the accuracy and the VGG-16 clearly outperforms the MLP, since the training accuracy of the VGG-16 starts to float between 95% and 100% before the twentieth epoch, while the training accuracy of the MLP reaches the 85% neighborhood after 60 epochs. The test accuracy is not as high as the training accuracy for both the VGG-16 and the MLP, however the mean of the VGG-16 at the end of the simulation is 81.29%, while the mean of the MLP test accuracy is 55.43%. The next evaluation compares the trends of the loss functions characterizing both models on the training set. The loss function used for both models is the *categorical cross entropy*. The VGG-16 mean of the values of the training loss function is a little less than 0.6 while the corresponding MLP mean is 1.6. Comparing the classification reports (see Table I for the VGG-16 performance and Table II for the MLP performance) of the two models it is clear that the VGG-16

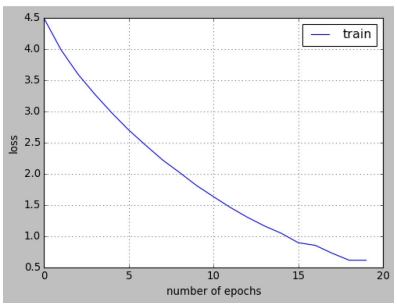

Fig. 3. Training loss

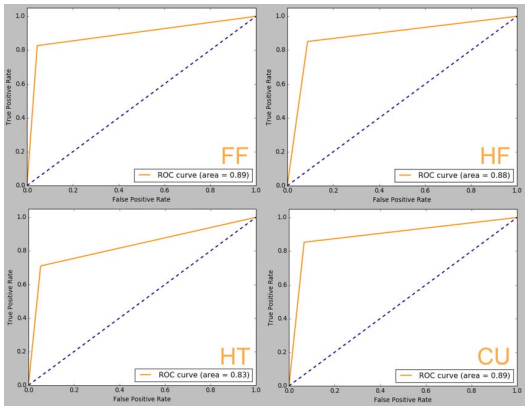

Fig. 4. Roc curves of the VGG-16

performs better by far.

The next comparison is between the VGG-16 and a generic CNN model. The generic convolutional neural network built for this study has two convolutional layers, the first one applies 64 filters, while the second one applies 128 filters. Each convolutional layer is followed by a pooling layer that performs a max pooling operation. The convolution filters have dimensions of 3x3 pixels, while the pooling windows have a size of 2x2 pixels. The other layers implemented are the input layer, which has 43 200 nodes, which is the same number of features of every image (160x90x3), the output layer, which has 4 nodes, one for each class considered in this study, and two fully connected layers. The first fully connected layer has 128 nodes, while the second one has 64 nodes. Every layer uses the *ReLU* activation function with the exception of the output layer that uses the *softmax* function. The number of epochs on which the CNN was trained is 30, while the batch size was 10. As well as the VGG-16 also this CNN was trained through the cross validation technique, with a Kfold=50. The loss function used was the categorical cross entropy, while the optimizer was the stochastic gradient descent Also in this case, although the generic CNN performs better than the MLP, the difference between the CNN performance and the VGG-16 is substantial. For what concerns the values of the loss function both models reach values floating around 0.6, however the CNN accuracy is much worse than the accuracy of the VGG-16. The value of the average accuracy of the generic CNN on the test sets reaches the value of 65.44% a value that pales compared to the 81.29% obtained by the VGG-16. It is interesting to notice that all of the models used have issues in identifying half torsos. Such result however is not surprising considering the fact the half torso is a class between the half figure and the close up, and in some cases they can be misclassified by humans too. Figure 4 shows the Roc curves for the VGG-16 model. As the reader can see the area under the *half torso* Roc curve is the smallest.

TABLE II
MLP CLASSIFICATION REPORT

	precision	recall	f1-score	support
Full Figure	0.60	0.63	0.61	15
Half Figure	0.66	0.57	0.61	15
Half Torso	0.48	0.48	0.48	15
Close Up	0.57	0.54	0.55	15
Accuracy			0.55	60
Macro Avg	0.58	0.55	0.56	60
Weighted Avg	0.58	0.55	0.56	60

TABLE III
GENERIC CNN CLASSIFICATION REPORT

	precision	recall	f1-score	support
Full Figure	0.74	0.75	0.75	15
Half Figure	0.71	0.62	0.66	15
Half Torso	0.55	0.59	0.57	15
Close Up	0.68	0.67	0.68	15
Accuracy			0.65	60
Macro Avg	0.67	0.65	0.66	60
Weighted Avg	0.67	0.65	0.66	60

IV. RELATED WORK

Only a few studies [9]–[11] centered on possible interactions between the shot classification and machine learning have been made in the past years While authors in [9] addressed the classification of images shot types as *long shots, medium shots* and *close ups*, in [10] a specific type of cinematographic shot, the *Over-the-shoulder* has been targeted. A large number of shot classes has been considered in the study presented in [11] where images are classified as one of the seven shot types: *extreme long shot, long shot, medium long shot, medium shot, medium close up, close up* and *extreme close up*. Since in all of these types of shot there is the human figure they used as discriminants the head size and its position in order to decide which shot is what. To this aim a semi-automatic face tracker is used to estimate the head size with respect to the whole image and its position. These two features were then fed to an SVM that classified the images into cinematographic shot types. The work proposed in this paper differs from those just introduced mainly for the methodology used. None of the cited papers used a fine tuned VGG-16 to classify the images into cinematographic shot classes. Most importantly the method proposed in this paper classifies the images automatically by itself, it does not need external interventions from humans or other algorithms. Thanks to its autonomy it would be more helpful to users who are outsiders to the world of machine learning.

V. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this paper we have used a fine tuned VGG-16 to address the cinematographic shot classification and we have performed different experiments on two real datasets. The results obtained so far are promising and we believe that they can be significantly improved by expanding the datasets under analysis in order to have more images for each class and also images related to all the cinematographic shot classes. The proposed approach could be also extended by integrating other machine learning techniques, such as feature boosting [12] and semantic segmentation [13]. We also intend to implement visual explanation techniques such as LIME [14] or Ebano [15] in order to understand which parts of an image have the most relevant influence on the network predictions.

REFERENCES

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[2] Lia Morra, Nunzia Coccia, and Tania Cerquitelli. Optimization of computer aided detection systems: An evolutionary approach. *Expert Syst. Appl.*, 100:145–156, 2018.

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[4] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. *A Survey on Deep Transfer Learning: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III*, pages 270–279. 10 2018.

[5] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[7] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[8] Jiawei Han, Micheline Kamber, and Jian Pei. 8 - classification: Basic concepts. In Jiawei Han, Micheline Kamber, and Jian Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 327 – 391. Morgan Kaufmann, Boston, third edition edition, 2012.

[9] Luca Canini, Sergio Benini, and Riccardo Leonardi. Classifying cinematographic shot types. *Multimedia Tools and Applications*, 62:51–73, 2011.

[10] M. Svanera, S. Benini, N. Adami, R. Leonardi, and A. B. Kovács. Over-the-shoulder shot detection in art films. In *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2015.

[11] I. Cherif, V. Solachidis, and I. Pitas. Shot type identification of movie content. In *2007 9th International Symposium on Signal Processing and Its Applications*, pages 1–4, 2007.

[12] P. Saxena, D. Saxena, X. Nie, A. Helmers, N. Ramachandran, N. Sakib, and S. Ahamed. Feature boosting in natural image classification. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 61–67, 2019.

[13] Z. Dong, R. Zhang, X. Shao, and H. Zhou. Multi-scale discriminative location-aware network for few-shot semantic segmentation. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 42–47, 2019.

[14] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[15] Francesco Ventura, Tania Cerquitelli, and Francesco Giacalone. Black-box model explained through an assessment of its interpretable features. In *New Trends in Databases and Information Systems - ADBIS 2018 Short Papers and Workshops, AI*QA, BIGPMED, CSACDB, M2U, BigDataMAPS, ISTREND, DC, Budapest, Hungary, September, 2-5, 2018, Proceedings*, pages 138–149, 2018.