Deep Learning based Beat Event Detection in Action Movie Franchises

N. Ejaz, U. A. Khan, M. A. Martínez-del-Amor, H. Sparenberg
Moving Picture Technologies, Fraunhofer Institute for Integrated Circuits (IIS)
Am Wolfsmantel 33, 91058 Erlangen, Germany.
{ejaznd, khanur, miguel.martinez, heiko.sparenberg}@iis.fraunhofer.de

ABSTRACT

Automatic understanding and interpretation of movies can be used in a variety of ways to semantically manage the massive volumes of movies data. “Action Movie Franchises” dataset is a collection of twenty Hollywood action movies from five famous franchises with ground truth annotations at shot and beat level of each movie. In this dataset, the annotations are provided for eleven semantic beat categories. In this work, we propose a deep learning based method to classify shots and beat-events on this dataset. The training dataset for each of the eleven beat categories is developed and then a Convolution Neural Network is trained. After finding the shot boundaries, key frames are extracted for each shot and then three classification labels are assigned to each key frame. The classification labels for each of the key frames in a particular shot are then used to assign a unique label to each shot. A simple sliding window based method is then used to group adjacent shots having the same label in order to find a particular beat event. The results of beat event classification are presented based on criteria of precision, recall, and F-measure. The results are compared with the existing technique and significant improvements are recorded.

Keywords: Shot Classification, Scene Recognition, Movies Classification, Beat Events Classification, Convolution Neural Network.

1. INTRODUCTION

Thousands of movies are being produced every year and thus automatic understanding movie contents is important for movie recommendation systems, efficient archiving and retrieval, content censorship, and scene driven retrieval systems etc. Many powerful features exist in the literature for action, event and activities recognition in the videos that combined multiple modalities including visual, text, audio, and motion [1,2,3].

A movie can be temporally sub-divided into a hierarchy of acts, scenes, shots, and frames. In a movie, a “beat” refers to the punctual changes in the storyline [4]. The frames of a video are easily available and efficient techniques for shot detection exist in the literature [4]. However, grouping shots into scenes is a much harder problem. Similarly, from the state of art in computer vision, detecting the semantic level of beats and acts is not possible. Popatev et al. [4] addressed the problem of detecting the beat events which are a combination of consecutive shots having the same event. The scheme computed high dimensional shot descriptors for various audio and visual channels including dense SIFT, CNN features, motion descriptors, audio descriptors, and face descriptors. The shots were then classified using Support vector machines. Finally, neighboring shots are grouped together to classify beat events. The authors also built an annotated data set of Hollywood action movies called “Action Movie Franchises”. The data set consists of 20 action movies along with shot and beat-level annotation over a set of 11 beat categories.

In this paper, a deep learning based scheme for beat event detection on “Action Movie Franchises” dataset is proposed. An appropriate training dataset is constructed for each of the beat categories after adjustment to some of the categories. Transfer learning is then used to modify and train the final layer of Inception-V3 [5] Convolutional Neural Network (CNN) using Softmax classification. For classification of shots in movies, firstly the shot boundaries are detected and a set of key frames are extracted for each of the shots. Considering a single shot at a time, CNN features are then computed for each key frame in the shot and top three class labels are predicted using the trained model. The top three class labels for each key frame are then integrated to yield a single label for the shot. In this way, a single event is associated with each shot in the movie. Finally, the consecutive shots with same event are grouped together to determine Beat Events. Despite not using audio, script and motion information, the proposed scheme outperforms the scheme of [4].

2. METHODOLOGY

2.1 Data Set Description

The "Action Movie Franchises" dataset [4] comprises 20 Hollywood action movies, each of which belongs to five famous franchises: Indiana Jones, Lethal Weapon, Die Hard, Rocky, and Rambo. In the data set, each shot of a movie is tagged with a class label among 11 possible class labels. Apart from the shot level annotation, the dataset also provides beat event level annotation. The 11 possible class labels along with their semantic meaning are summarized in Table 1. During annotation of movies, some annotations are labeled as difficult if they are ambiguous or semantically hard to detect. The beat-event annotation covers about 60% of the total movie footage.

No.	Category Name	Category Explanation
1	Pursuit	Chasing in any form
2	Battle Preparation	Preparation, Training and Planning for Battle
3	Battle	All types of fights including physical fight, arrow shooting, war etc.
4	Romance	Romance between hero and heroine including physical intimacy and romantic dialogues
5	Despair Good	Feeling of loss of a hero (or any positive character)
6	Joy Bad	A villain (or a bad guy) expressing joy, usually after winning a fight
7	Good Argue Good	A dialog between two (or more than two) good characters
8	Good Argue Bad	A dialog between a good character and a bad character.
9	Bad Argue Bad	A dialog between two (or more than two) bad characters
10	Victory Good	A temporary victory of good character(s)
11	Victory Bad	A temporary victory of bad character(s)

Table 1: Categories of Action Movie Franchises dataset

2.2 Dividing some classes into sub-classes

There are some really broad beat categories in "Action Movie Franchises" dataset. For instance, the class Battle covers all aspects of fighting including physical war, sword war, boxing, wrestling, battle, wars, etc. Because of this broad semantic associated with such large classes, a large intra-class variation among them is expected. In order to cope up with this intra-class variation, some of the classes are sub-divided into further sub-classes as shown in Table 2. For the sake of classification, any label with the sub-class is then classified as the major class.

No.	Category	Sub-Categories
1	Battle	Physical War, War Destruction, Bomb Explosion, Fire Attack, Arrows' War
2	Pursuit	Car Pursuit, Bike Pursuit, Horse Pursuit, Men Pursuit
3	Despair	Despair by Expressions, Despair in Jail

Table 2: Defining Sub-categories to reduce intra-class variation

2.3 Narrowing some classes

Some of the classes are narrowed down in the first level of classification. Instead of having separate classes for Good Argue Good, Good Argue Bad and Bad Argue Bad, we have only one category "Argument". Similarly, we have broad classes of Joy, Despair, and Victory instead of Joy Bad, Despair Good, Victory Good and Victory Bad. Once the classification result is Argument, Despair, Joy, or Victory, further classification into narrower categories was done using a separate classifier (as will be discussed in Section 2.5.2). In this way, we have a total of 16 beat categories: Physical War, War Destruction, Bomb Explosion, Fire Attack, Arrows' War, Car Pursuit, Bike Pursuit, Horse Pursuit, Men Pursuit, Despair by Expressions, Despair in Jail, Romance, Despair, Joy, Argument, and Victory.

2.4 Training

For each of the 16 beat categories, a dataset is prepared by collecting relevant frames from various sources including Youtube videos, movie trailers, and full-length movies. For each category, about 750 images are collected.

Convolutional Neural Networks (CNNs) are a class of deep learning models which are designed to simulate the visual signal processing in central nervous systems [6]. Recently, CNNs gained popularity especially for computer vision related tasks [7,8,9]. A key challenge in applying CNNs to beat event detection in movies is that there are no labeled training samples for the identified categories. The collection of huge data set is a daunting task. In order to overcome this difficulty and develop a universal representation, we employ transfer learning to transfer knowledge from labeled image data that are problem-independent. Transfer learning is a popular machine learning technique in which the information obtained while training one machine learning task can be utilized for another task especially if two tasks are related to one another [16]. The prime advantage of transfer learning is that it is helpful to avoid starting from scratch for training. This is a huge advantage as training from scratch needs huge computational resources, long waiting time, and huge training data. This approach of transfer learning has already yielded superior performance on various image recognition tasks [10,11].

The task of beat event classification is closely related to that of image classification. Therefore, we use Inception-V3 CNN which was trained on the ImageNet data that contains millions of labeled images with thousands of categories. In our work, the last layer of Inception-V3 pre-trained model is removed and a new layer is added to train on our dataset. In this way, the pre-trained model is then used directly as feature extractor to compute representations. A dropout layer is added as a so-called penultimate layer whose task is to randomly drop the output of 50% neurons. This is being done to reduce over-fitting and increase generalization [12]. After applying dropout, ReLU (Rectified Linear Unit) activation function is applied for introducing nonlinearity in the training, so that the network may generalize well for the unseen data. For each input image, the output of the penultimate layer is found as:

$$y_m = ReLU\left[\sum_n W_{m,n} x_n + b_m\right] \tag{1}$$

where W_m represents the weights of the neurons and b_m is the bias for the mth beat event. The notation x_n represents the nth pixel of the input image. In order to get the probabilities of all beat event classes, the output of the penultimate layer is given to Softmax classification layer to get probability distributions.

$$p_m = \frac{e^{y_m}}{\sum_{i=1}^{n} e^{y_i}} \tag{2}$$

p_m represents the probability of mth beat event in the set of 16 beat events. A cross-entropy function is used to calculate the error between the estimated distribution and the true distribution. The Softmax classifier minimizes the error between the estimated distribution and the true distribution. The training is run for 50,000 iterations and the training and validation batch sizes are kept to 500 with a small learning rate of 0.005. 80% images are used as training images, 10% images are used as validation images and 10% images are used as test images. The test accuracy of the model is 82.5% and the training/validation margin is minimal for training/validation accuracy and cross-entropy error during training.

2.5 Shot and Beat Event Classification

Figure 1 shows the major steps in shot classification. The first step for shot classification is to find shot boundaries. The shot boundaries for “Action Movies Franchises” dataset are given and are used as it is.

2.5.1 Key Frame Extraction and Classification

Considering one shot at a time, the next task is to find key frames for a particular shot. The key frames are representative or salient frames which best describe the contents of the shot. We utilize a simple technique for key frame extraction for two reasons. Firstly, to save computational time, and secondly, because of the observation that the overall performance of the system is not likely to be heavily dependent on the quality of key frames. We use the technique of [13] with some modifications. Instead of using all of the proposed features of [13], we use only color histogram in HSV color space. After obtaining the histogram of a particular frame in the shot, a color quantization step is applied to reduce data into 16 bins for hue component, and 8 bins for each of the saturation and intensity components. All three histograms are then normalized in the range of [0-1]. The three histograms are then combined to make an aggregated histogram vector of size 32. From the shot under consideration, the histogram is computed for each 20^{th} frame. The difference of histograms of neighboring frames is computed and a new key frame is declared if a significant difference is found. For each detected key frame, its standard deviation is computed. If the standard deviation is very low, the frame is discarded. This is done in order to prevent useless frames that may include totally black frames, totally white frames, and faded frames etc.[13]. Each key frame is then classified using the trained model. Instead of assigning only one class for each key frame, we pick top three classes corresponding to top three probabilities.

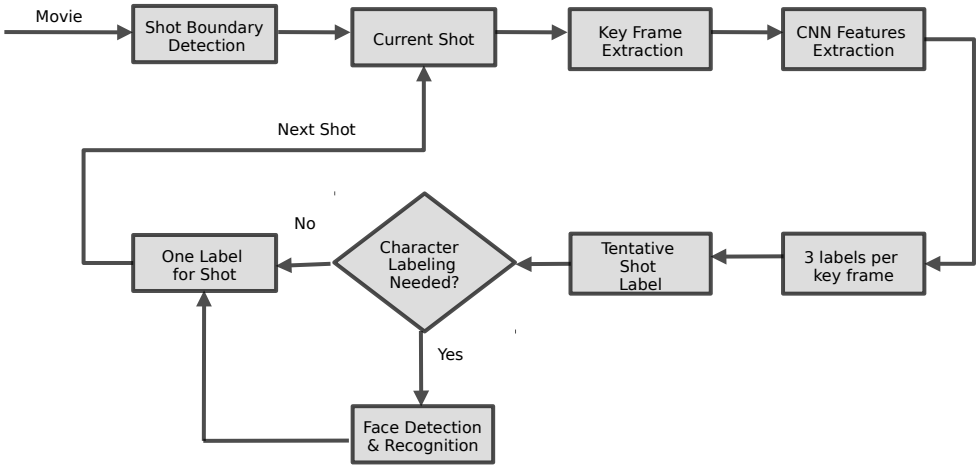

Figure 1: Process for Shot Classification

2.5.2 Classification Refinement

For some of the beat categories, further frame analysis is needed based on additional information from the movies. These categories include Good Argue Good, Good Argue Bad, Bad Argue Bad, Despair Good, Joy Bad, Victory Good and Victory Bad. If the classification result is one of the Argument, Despair, Joy, or Victory, further steps are taken to classify the involved characters. For this purpose, the classification of appeared characters into good or bad is necessary. For differentiating between good and bad characters, a database of images is developed by collecting images for all the prominent characters in the 20 movies of the dataset. On average, 5 major characters per movie are selected and images corresponding to those characters are collected and then used to train a facial recognition network [14]. While making this training dataset, each character is manually labeled as “Good” or “Bad”. A two-step approach is then used. In the first step, the humans are detected for a single key frame in the shot (the central key frame). For the purpose of human detection, a regression based network YOLO [15] is used to detect humans. Using the bounding box information provided by YOLO, the humans are then cropped from rest of the image. Secondly, the detected humans are classified into respective characters using the trained network. Next, if the original classification is Argument, then a further narrow classification into Good Argue Good, Good Argue Bad and Bad Argue Bad based on majority of respective characters appearing in the central key frame. Similarly, Despair, Joy and Victory are further classified into Despair Good, Joy Bad, Victory Good and Victory Bad.

2.5.3 Shot Classification

For a single shot, we have a set of key frames and three class labels corresponding to each key frame with their relative probabilities. The next task is to integrate this information to assign a single label to the shot. This is done in following

steps: (i) Find the number of occurrences of each unique label, (ii) Sum up the relative probabilities of each unique label, (iii) Multiply the number of occurrence of each label with the sum of probabilities to get a list of weighted probabilities (iv) Assign a single label to the shot by selecting the label which has the highest relative weight.

2.5.4 Beat Event Classification

After a single class label has been assigned to each of the shots in the movie, the next task is to group the neighboring shots with the same labels to get the beats. Instead of simply grouping the shots as done in [4], we used a simple sliding window based approach. A window of size 3x1 is placed on neighboring shots starting from the first shot. There are two possibilities: (i) If each label is unique, then declare each shot as a separate beat event, (ii) If each label is not unique, then select the majority label as beat event label for the complete window. The process is repeated with non-overlapping windows of equal size. If two adjacent windows have the same label, then they are jointly given one beat label. The window size is kept to be small because a larger window size may induce unnecessary overlapping errors.

3. EXPERIMENTS AND RESULTS

The experimental setup included a GPU GeForce GTX 1050 Ti (768 cores, 4GB GDDR5). TensorFlow 0.12 was used as a deep learning platform and the implementation was done in Python 2.7 and OvenCV 3.0. With this experimental configuration, the average time for classification is 20 FPS for a 720p movie.

Table 3 shows the comparative results of our technique and that of Popatev et al.[4] for beat event detection on Action Movies Franchises dataset. For the proposed scheme, we computed precision, recall, and F-measure. For Popatev et al. [4], only precision values were available. In [4] training was done as “leave 4 movies out” which means that 16 movies from the dataset were used as training and 4 movies were used as test data. The movies in a particular franchise are usually somehow related and thus the presence of related movies in training data is likely to affect the tests. In our case, utmost care was taken that the training images do not include any frames from the test movies. The average precision reported in [4] for beat event classification on action franchises dataset was 0.17 which is significantly lower than the results of the proposed scheme.

From Table 3, it is evident that all those categories which are semantically clear and less ambiguous exhibited good results. Table 4 shows the confusion matrix to visualize the classification performance. The results of class Battle are satisfactory primarily because of reduction in intra-class variance by dividing battle into multiple categories. The confusion matrix in Table 4 shows that battle class mostly gets mixed with Pursuit category and thus results in misclassification. This is the natural outcome as Battle scenes are mostly associated with Pursuit scenes where characters are found chasing each other.

	Battle	Good Argue Good	Good Argue Bad	Romance	Despair Good	Preparation	Pursuit	Joy Bad	Bad Argue Bad	Victory Bad	Victory Good	Mean
Precision [4]	0.39	0.17	0.12	0.23	0.06	0.27	0.35	0.05	0.06	0.04	0.15	0.17
Precision(Proposed)	0.64	0.81	0.59	0.69	0.89	0.86	0.63	0.58	0.58	1.00	0.92	0.74
Recall(Proposed)	0.91	0.81	0.86	0.70	0.49	0.06	0.82	0.71	0.58	0.33	0.26	0.59
F-measure(Proposed)	0.76	0.81	0.70	0.70	0.63	0.10	0.71	0.64	0.58	0.50	0.41	0.59

Table 3: Performance Comparison for Beat Event Classification

The results also show that for the categories involving arguments, the accuracy is much superior to the existing scheme [4]. Again, this performance is achieved because the argument class is initially separated as a single category and is further fine tuned based on face detection and recognition. Table 4 reveals that Good Argue Good mostly gets mixed with Romance and Pursuit. For the case of Romance, sometimes it is difficult to differentiate an argument between two characters as having argument or romance. Without the support of additional modalities like audio and text and absence of temporal features, it is sometimes difficult to differentiate between Good Argue Good and Romance especially if the characters are having an argument when they are physically close to each other. The deep learning based face recognition is efficient, but the training was not done on each and every character of the movie. For this reason, we have some mis-

classifications in character recognition which ultimately affects the decisions related to Good Argue Good, Good Argue Bad and Bad Argue Bad. The class Good Argue Bad is sometimes confused with Joy Bad which is a closely associated category. The Romance category shows reasonably good performance but frequently gets mixed with Good Argue Good when the characters are having a romantic talk involving no physical intimacy and visual features conceived by mere images are unable to catch the semantics of romance. Despair Good is a frequently occurring beat event in almost all of the action movies. The semantics covering Despair Good has much broader contextual meanings which include expressions of despair, hopelessness and dejection. As explained, the Despair category is also broken into sub-categories to better capture the semantic variation associated with this category. Despair Good class is still closely linked with other some of the other classes. For instance, as the Table 4 suggests, the Despair Good is mostly confused with Battle, Pursuit, and Joy bad. Considering that the despair situation of a good character mostly happens during a state of fighting and war, the Despair Good scene in that case comes with a battle like environment with weapons, ambiance and/or dressing etc. suggesting a battle. Thus the confusion of Despair Good with Battle is natural. However, the sub-categorization of Despair class has helped to achieve comparatively better accuracy. As described earlier, Pursuit is also somehow related to many battle situations and thus it is a natural extension that Despair Good situations are mixed with the Pursuit class.

A particularly interesting fact is that Despair Good also get mixed with Joy Bad on fairly reasonably occasions. The natural explanation is that the despair of a good character generally results in a joy for the bad character. Since only a single beat event label was assigned to a beat, the algorithm confuses between these related categories. The biggest failure of our technique is in classification of beat event Battle Preparation which is significantly confounded with Battle, Good Argue Good and Pursuit. The confusion of Battle Preparation with Battle and Pursuit is natural as Battle Preparation scenes mostly involve people practicing the use of weapons or doing physical practice of a battle. In most of the situations, the trained network fails miserably to distinguish between a real battle and a battle for practice. Another form of Battle Preparation in the dataset is which should be more clearly labeled as Battle Planning- a group of people discussing their battle plans and tactics. This type of Battle Preparation gets easily confused with any of the argument classes which mostly happen to be Good Argue Good in our case. More work is needed for the class Battle Preparation which may consider either adding additional modalities or improving training dataset. The Pursuit category has the most confusion with the class Battle for the reasons already explained. The Joy Bad class has a remarkable boost over Popatov's scheme [4] but gets confused with Good Argue Bad as in most of the scenes the Joy bad occurs when having an argument with a good character. The Bad Argue Bad argue has few scenes in the ground truth. Finally, two categories related to victories (Victory Good and Victory Bad) have low accuracies, by mostly getting confused with the events of Battle. The victories, as defined in the ground truth, also include temporary victories during Battle where an opponent is temporarily down. Thus, the victory events are difficult to be differentiated from Battle unless they are explicitly celebrated.

	Battle	Good Argue Good	Good Argue Bad	Romance	Despair Good	Preparation	Pursuit	Joy Bad	Bad Argue Bad	Victory Bad	Victory Good
Battle	354	2	1	4	3	0	22	2	0	0	0
Good Argue Good	5	237	10	13	4	0	18	5	0	0	0
Good Argue Bad	3	0	120	1	3	0	5	6	1	0	0
Romance	1	19	0	69	3	0	2	2	2	0	0
Despair Good	66	10	33	8	186	0	35	40	3	0	0
Preparation	47	23	4	3	5	6	16	3	1	0	1
Pursuit	30	0	5	0	2	0	168	0	0	0	0
Joy Bad	3	0	26	1	1	0	0	84	3	0	0
Bad Argue Bad	6	0	1	1	1	0	1	0	14	0	0
Victory Bad	6	0	2	0	1	1	0	2	0	6	0
Victory Good	28	2	3	0	0	0	1	0	0	0	12

Table 4: Confusion Matrix for Beat Event classification

4. CONCLUSION

In this paper, a deep learning based technique for shot classification and beat event detection in movies is presented. The proposed technique is generic in nature but the dataset was developed for running it on “Action Movie Franchises” dataset and for detection of 11 beat events identified in this dataset. The proposed technique outperformed the state of the art on this dataset by quite some margin for beat event detection. The proposed technique is flexible for adding new beat categories and can be easily tailored to other genres of movies. For some of the beat events (for instance Battle Preparation), the performance measure is below par and need significant improvements. This low performance for some beat events is the result of intra-class similarity with other beat events. In future, we intend to use additional modalities including semantic audio analysis and text analysis for improving the performance. Moreover, we aim to extend our framework for additional video genres and an extended set of beat events.

5. REFERENCES

[1] C. Li, Z. Huang, Y. Yang, J. Cao, X. Sun and H. T. Shen, "Hierarchical Latent Concept Discovery for Video Event Detection," in *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2149-2162, May 2017.

[2] Z. Xu, Y. Yang, A.G. Hauptmann, “A Discriminative CNN Video Representation for Event Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1798-1807, 2015.

[3] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, “Finding Actors and Actions in Movies”. *ICCV 2013 - IEEE International Conference on Computer Vision*, Dec 2013, Sydney, Australia. IEEE, pp.2280-2287, 2013.

[4] D. Potapov, M. Douze, J. Revaud, Z. Harchaoui, C. Schmid. “Beat-event Detection in Action Movie Franchises”, arXiv preprint arXiv:1508.03755, 2015.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 2818-2826.

[6] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798–1828, 2013.

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

[8] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE conference on computer vision and patternrecognition, pages 4694–4702, 2015.

[9] T. Chen, L. Lin, L. Liu, X. Luo and X. Li, "DISC: Deep Image Saliency Computing via Progressive Representation Learning," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1135-1149, June 2016.

[10] M. Oquab, I. Laptev, L. Bottou, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[11] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, pages 818–833. Springer, 2014.

[12] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[13] N. Ejaz, T.B. Tariq, S.W. Baik, Adaptive key frame extraction for video summarization using an aggregation mechanism, Journal of Visual Communication and Image Representation, 23 (7) (2012), pp. 1031-1040.

[14] F. Schroff, D. Kalenichenko, J. PhilbinS,FaceNet: A Unified Embedding for Face Recognition and Clustering,IEEE Conference on Computer Vision and Pattern Recognition (CVPR)}, June 2015.

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks?. In Advances in neural information processing systems, pp. 3320-3328, 2014.