Sports Video Classification using Objects as Attributes

Amanpreet Walia
amanpreet.walia@mail.mcgill.ca

Boshra Badran
Boshra.Badran@mail.mcgill.ca

December 11, 2018

Abstract

Although robust low-level image features have proven to be effective representations for a variety of high-level visual recognition tasks, as the visual recognition tasks become more challenging, the semantic gap between low-level feature representation and scene description increases. However, previous work has shown that using objects as feature representation for a complex scene can provide better representation for scene classification than using low-level image features like interest points in an image. In this project, we try to carry forward this idea by using detected object features to classify video scenes. We have chosen 8 video categories based on Sports Events dataset and collected videos from online Video Database like YouTube to construct the dataset, and applied various techniques to classify the video scenes which yielded interesting results.

1 Introduction

Scene classification is a fundamental challenge to the goal of automated visual perception. Although humans are proficient at perceiving and understanding scenes, making computers do the same poses a challenge due to the wide range of variations in scene appearance. Much of the progress in machine vision related to this area is reliant on low level features in an image(or a scene) such as SIFT [1], filter banks [1], GIST [1], etc. Using sophisticated statistical models, we have achieved good successes in high-level recognition tasks e.g. object detection. However, as the semantic gap between low level features (e.g pixels) and high level tasks (e.g.scene recognition) increases, a model has to do more and more work towards achieving high-level goals. This task becomes even more difficult when the variation of scene information is not just spatial but temporal as well. In this project, we tried to address these two problems while at the same time working at a higher semantic level for scene understanding.

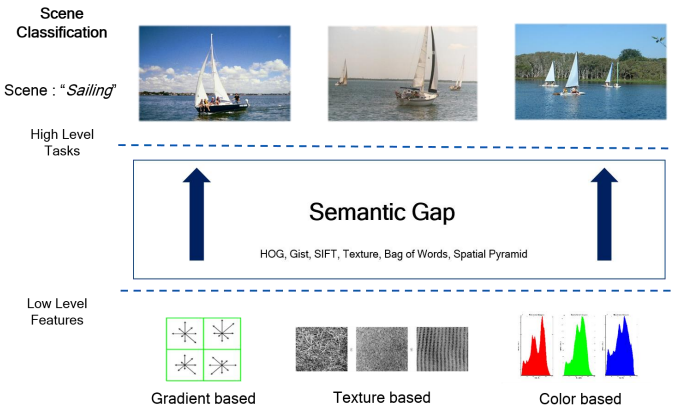

Figure 1: Semantic gap between low level features and high level tasks

To solve the problem of temporal variation, we have used the concept of *Keyframes*. Keyframes are representative frames of a video stream, that provide a compact summary of the video content. In this project, we propose a representation of video scene based on *objects* present in the fixed number of these extracted *Keyframes*. To accomplish this, we describe video scenes by collecting the responses of many object detectors on each extracted keyframe of a video and combined the collected features for each keyframe to form a representation for the whole video. By using a large number of such object filters, our object bank representation can provide rich information about a scene which as shown in results will be suitable for video scene classification.

Motivated by the good results obtained from using a limited number of object detectors, we used a pre-trained VGG16 network trained on 1000 objects and used this network to extract output of last second layer

as a feature for each keyframe scene of the video. This approach improved the results significantly as compared to just using object bank limited to a few (177) object detectors as already predicted in [1].

2 Background

2.1 Object Bank

Given an image, an object filter response can be viewed as the response of a "generalized object convolution" where we take an object template, and scan it across the image, resulting in a map of face filter responses. We first run a large number of object detectors at multiple scales. For each object at each scale, use a three-level spatial pyramid representation of the resulting object filter map, resulting in `No.Objects`×`No.Scales`×$(1^2 + 2^2 + 4^2)$ grids. An object filter descriptor of an image is a concatenation of features described in each of these grids. We compute the maximum response value of each object, resulting in a feature vector of No.Objects length for each grid[1].For this project, we have used 177 objects detectors at 12 detection scales and 3 spatial pyramid levels resulting object bank with a dimension of `44604` dimensional feature vector. As suggested in paper[1], Zipf's law trend suggests that a small proportion of objects occurs much more frequently than the majority, therefore the assumption to use just 177 object detectors obtained from Imagenet[5] becomes justified.

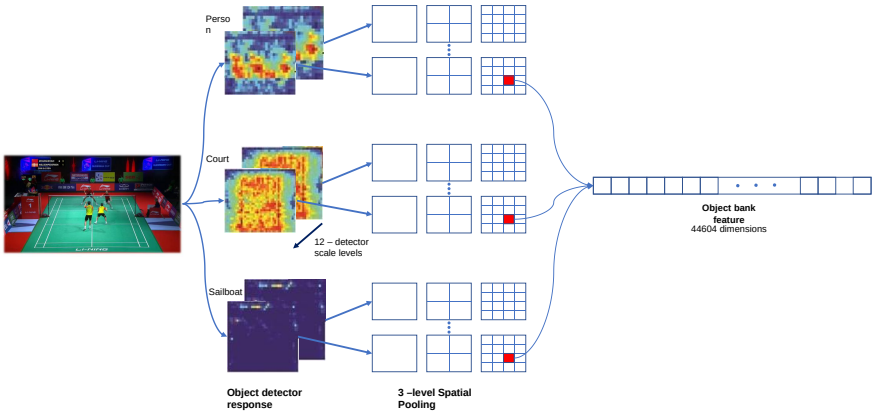

Figure 2: Object bank Construction

2.2 Pre-trained VGG16 Neural Network

VGG-16 is a convolutional neural network that is trained on more than a million images from the ImageNet database [5]. The network is 16 layers deep and can classify images into 1000 object categories. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

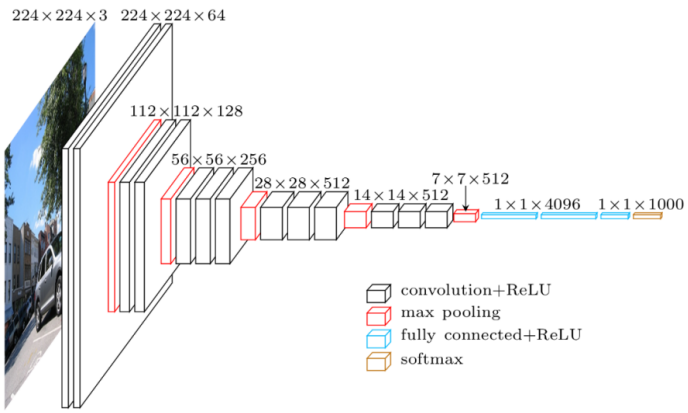

Figure 3: VGG 16 Macroarchitecture [6]

3 Data Collection & Description

For this project, we focused on classifying sports videos and utilized following existing datasets :

- **UIUC Sports Event Dataset[2]**: This dataset contains 8 sports event categories: rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock climbing. This dataset is used in [1] to classify the sports scenes in images using object bank instead of low-level features in the paper[1] which is major motivation for our project. Therefor for video classification, we used the same sports categories as given in UIUC Sports Events Dataset considering that using object bank for image scenes has already yielded a very good classification accuracy.
- **Sports-1M Dataset[3]**: Sports 1M Dataset has links to 1,133,158 YouTube videos for 478 sports categories. This was useful in getting videos for the 8 sports categories we are considering for this project. We used the given links for respective categories to look for the most informative segment of the video which can represent the video accurately.

3.1 Our Dataset

We built the dataset for this project by choosing 40 videos per sport category. Therefore, the resulting dataset consists of 320 high quality-videos equally divided among 8 sports categories.

Dataset is available at: https://github.com/amanwalia92/SportsVideoClassification/tree/master/Dataset/Videos

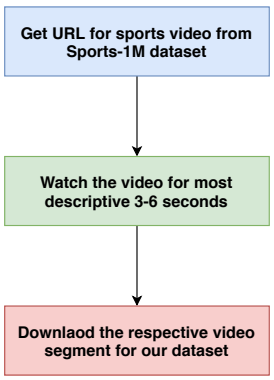

Figure 4: Pipeline to construct the Video Dataset

4 Methodology

4.1 Extract Keyframes

From the videos we collected in the dataset, we sample seven key frames from the video. To do this, we first converted each frame to LUV color space and then calculate the sum of absolute differences between all the pair of consecutive frames. To select the most important keyframes, we smooth the plot of differences of consecutive frames and then select the top 7 local maxima. Code for this algorithm can be found on : https://github.com/amanwalia92/KeyFramesExtraction

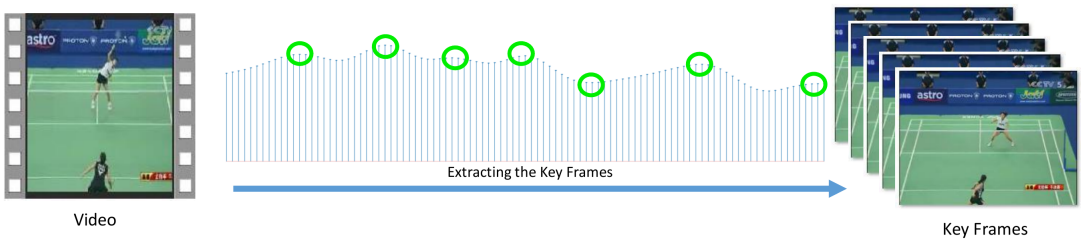

Figure 5: Extract Keyframes

4.2 Object Bank features

4.2.1 Feature Selection

Feature selection can be defined as the process of selecting the important features from existing ones. "The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data." [7]

Intuitively, a few key objects can discriminate a scene class from another, therefore we can select a subset of these features from object bank representation of each key frame for a video, without losing the discriminatory power, and hence feature selection will provide a more semantically meaningful compression. We investigate content-based compression of the high-dimensional OB representation that exploits raw feature, object, and (feature + object) sparsity, respectively. We used two common feature selection methods:

1. ***L1-based feature selection***

   Adding L1 penalty to a model leads to a sparse solution, which means it will set the coefficients for the features that are less contributed to the model to zero, thus selecting the features. Applying L1 feature selection to the video Object Bank feature vector reduces the dimension from 267624 features to just 1302 features! for each video which helps us in reducing the training time and the used memory, with only marginal sacrifice in accuracy.

2. ***Tree-based feature selection***

   Tree based feature selection depends on computing the importance of the features and then selects the most important features. We used Gini importance, which calculates the importance of the feature according to how many splits in the tree has this feature. After that, we set a threshold (1e-5), and the features which their importance is less than this threshold were discarded. Tree based feature selection reduces the dimension of the video Object Bank feature vector from 267624 features to 26070 features, about 10 times less.

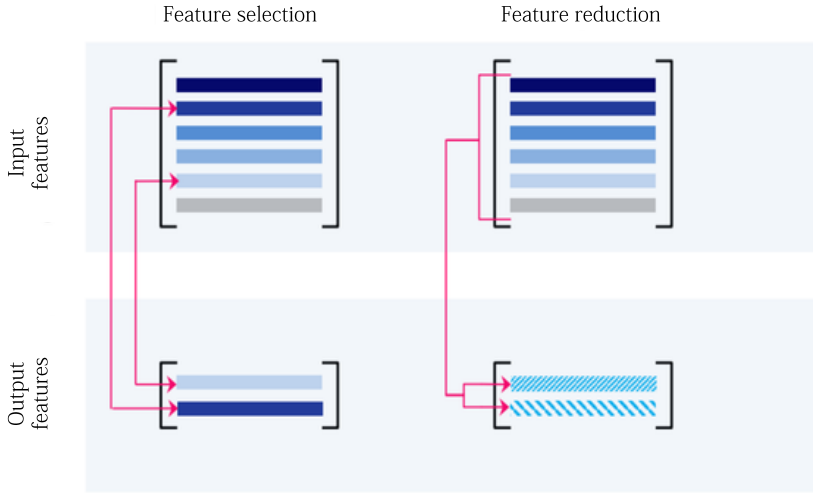

Figure 6: Feature Selection Vs. Feature Reduction

4.2.2 Feature Reduction

Feature reduction is another way that is used to reduce the dimension. The main difference between feature reduction and feature selection is that feature selection selects the important features and neglect the others, while feature reduction looks to all the features, and creates a new combination of them. We used two feature reduction techniques:

1. ***Principal Component Analysis (PCA)***

   Principal Component Analysis is a dimensionality reduction method that transforms the feature space to a new lower dimensional space determined by the principal component. It mainly depends on the variance

of the features, as the features that have big variance are the ones that will be used to build the new space. Using PCA reduces the features dimensional space from 267624 to new 11520 features, and we got the same accuracy as using the original number of features.

2. ***Random Projection***

   Random projection is another method for reducing the dimension. The main difference between PCA and random projection that the direction of projection for PCA depends on the features, while it is independent of them for random projection. In the project, we chose the Gaussian Random Projection which reduces the dimension by projecting the features space to a randomly generated matrix whose components are drawn from a Normal distribution with mean zero. Gaussian Random Projection reduces the features dimensional space from 267624 to new 38880 features. Using this we only get a marginal loss of accuracy.

Method	Number of features
L1	1302
Tree Based	26070
PCA	11520
Random Projection	38880

Table 1: Dimensional reduction methods and the new number of features

4.3 VGG16 features

4.3.1 Transfer Learning to extract features

Training an entire convolutional network is very difficult due to the constraints of time and resources, that's why we have used pretrained VGG-16 network and used the second last layer to extract features for each key frame of every video. As this neural network is trained on 1000 objects, this will improve the prediction accuracy as compared to using object bank filters where we have just 177 objects.

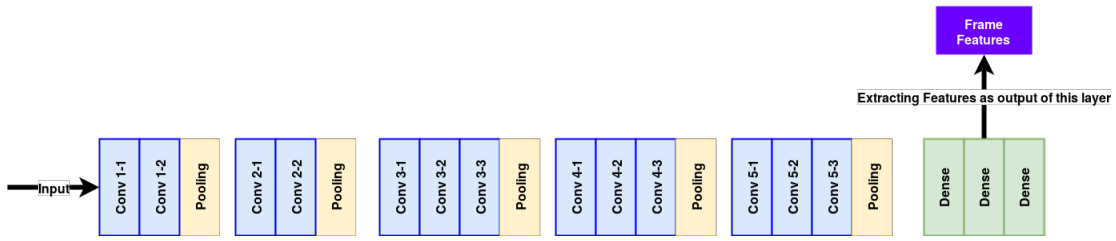

Figure 7: VGG16 Feature extraction

4.4 Video Level Features

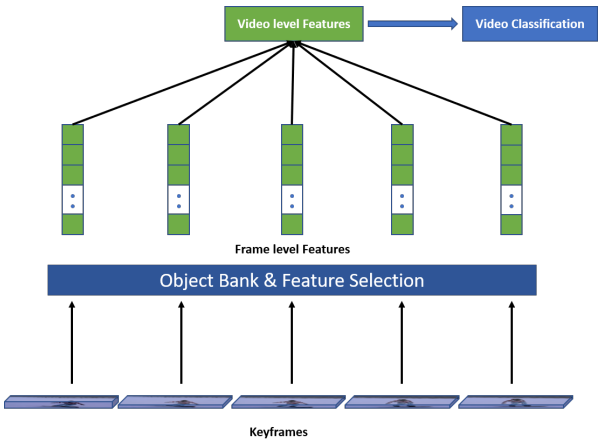

Figure 8: Video Features from Frame level Features

Once we extracted the features for each of the key frames using either the object bank (after feature selection or reduction) or VGG16 feature extraction technique, all the frame level features are concatenated together to form a video level feature. We use this video feature vector for training and classification of video[8].

4.5 Classifiers

After prepossessing the key frames, and concatenating them to one video feature vector, we will start learning the classifiers. We chose three major classification methods, Logistic Regression ,Support Vector Machine and Multi-Layered Perceptron. Logistic Regression and SVM techniques are used in the original paper[1] which motivated us to use them for the modification we proposed just to see how well it goes for it.MLP technique was introduced as novelty for VGG16 object features which yielded the best accuracy.

1. ***Logistic Regression:*** We used multinomial logistic regression as we have 8 classes. Given the features and the classes, Logistic regression will build a model that fits these data by assigning weights (coefficients) for each feature. We used L2 penalty for the model, and to choose the best regularizer parameter (Lambda) we used 5 folds cross validation. Then by giving the model a new video feature, it will be able to predict to which class does it belong.

2. ***Support Vector Machine:*** From the SVM family, we have used Linear Support vector machines for multi-class classifications. Since we have more than two classes, we used train one vs. rest classifiers. We chose SVM because the high-dimensional data in our dataset.

3. ***Multilayer Perceptron:*** Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function by training on a dataset. One advantage of using MLP is that it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. We have used this classification method only for VGG16 extracted video features where it provides superior performance compared to all other classification methods.

5 Results

5.1 Object Bank

We tested the above feature extraction and feature reduction methods, with the two classifiers, and we got the results as shown in the graph.

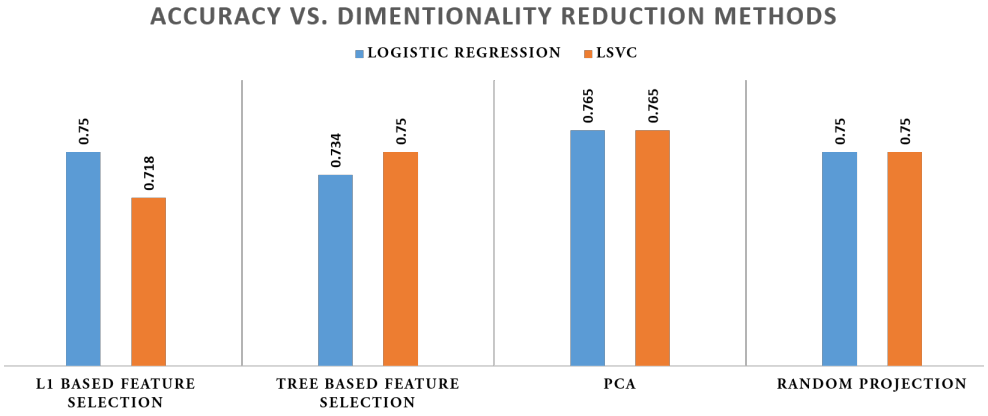

Figure 9: Accuracy when using dimentionality reduction methods for OB features with LR and LSVC classifiers

From the graph, the accuracy for all the methods was approximately the same, ranging between (0.71 and 0.76). The best accuracy was achieved when we used PCA.

We also compute the running time for each method, as shown in the table.

From the table, we can investigate that there is a proportional relationship between the number of the features and the running time, as the number of features increases we need more time to process these features. In addition, the running time for the Linear support vector classifier is always taking less than the running time for the logistic regression.

Method	Number of features	Classifier	Running time (seconds)
L1 based feature selection	1302	LR	8.96
		LSVC	1.63
Tree Based feature selection	4345	LR	198.72
		LSVC	64.46
PCA	11520	LR	67.57
		LSVC	5.17
Random projection	38880	LR	296.92
		LSVC	107.01

5.2 VGG16

For the VGG16 part, we just used the original feature vector for each video without doing any feature selection. We started with the Multi Layer Perceprtron classifier for the 4096 features extracted for each frame ,and tried to tune the hyper parameters to study their effects.

We concentrated on the following hyper parameters: type of optimizer, optimization function, and number of neurons in the hidden layer. We used cross-validation to choose the number of hidden neurons, and the number that gives a best accuracy for our dataset was 200.

For the type of optimizer and the activation function, we tried the combination between all of them (`Adam - SGD - lpfgs`) and (`identity - logistic - Tanh - Relu`) respectively, and we run the model for 20 iterations. The results that we obtain are : `lbfgs` with `logistic` as shown in the graph below.

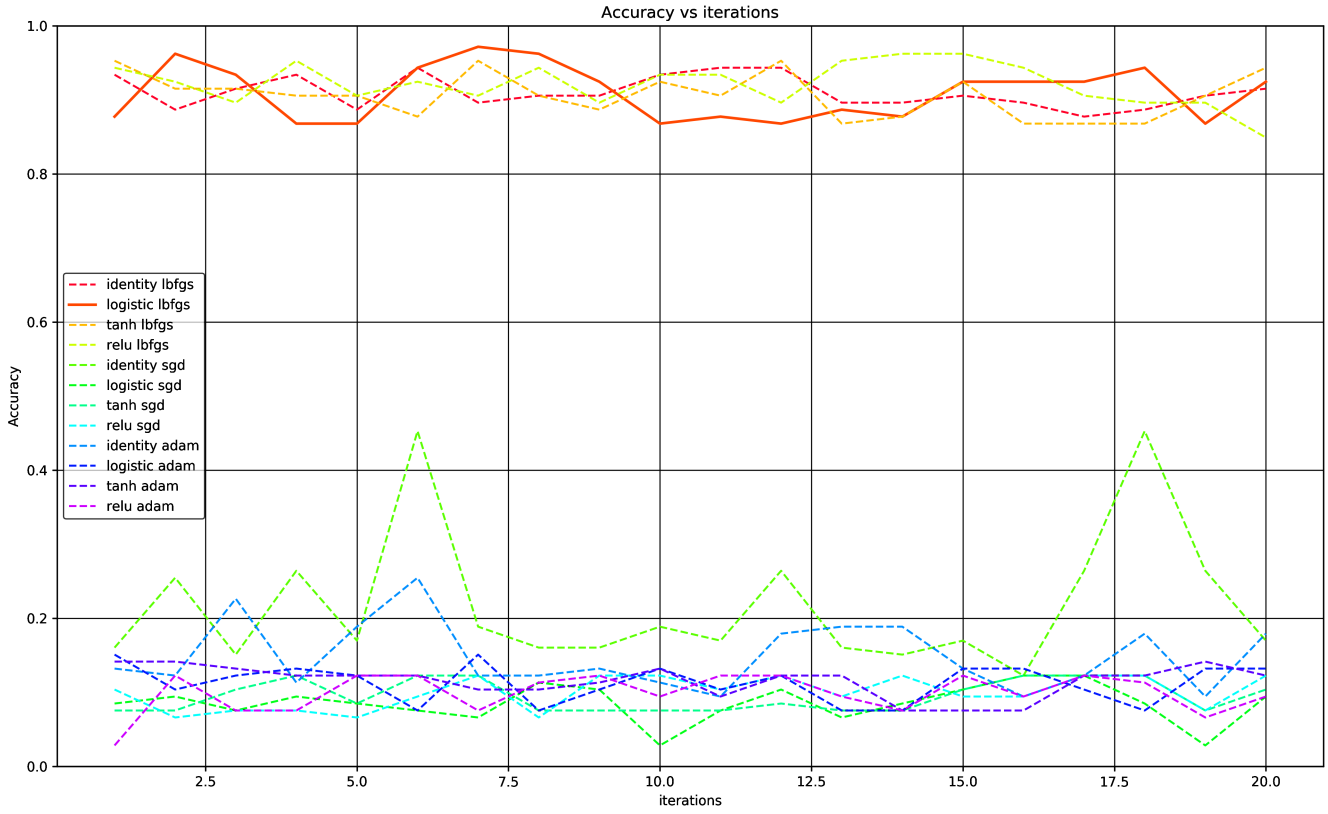

Figure 10: Accuracy of MLP with different optimizers and activation functions

From the graph, we can see that `SGD` and `ADAM` didn't work well with our data, and the average accuracy was around 0.1 for any activation function. While for `lbfgs`, it always gives a good accuracy with any activation function, but its response was the best with the logistic function. One reason we can justify this behavior of the optimizers is that Adam works well with large datasets, while for small datasets like ours, `lbfgs` works better (Our data sets contains 320 videos).

After that, We compared the accuracy of the three classifiers, Logistic regression, linear support vector classifier, and multi-layer perceptron classifier as shown below:

The bar chart shows that MLP with lpfgs optimizer and logistic function gave the best accuracy among the three classifiers.

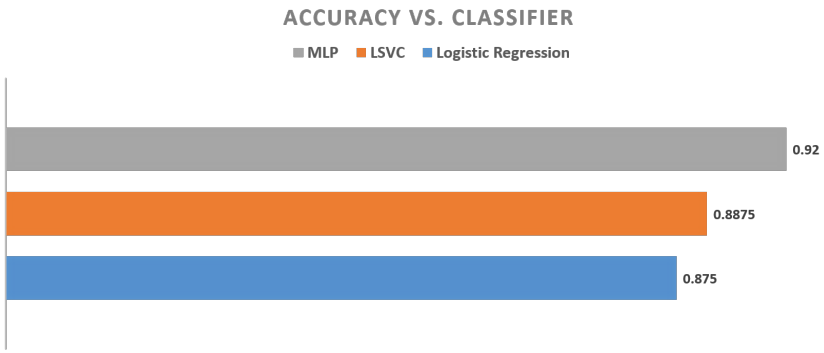

Figure 11: Accuracy when training LR, LSVC, and MLP on the resulting feature vector of VGG16

6 Conclusion

In this project, we tried to work on the idea of using objects to classify video scenes. We constructed our own dataset that contains 320 videos for 8 sports categories. We started by processing a video to extract its most prominent temporal information using key frames and then use two methods to extract the object level features from each frame. The first method uses Object Bank mentioned in [1], where we used 177 object detectors to construct feature vector for each frame by passing the frame through these object detectors, then combine the feature vectors for all the key frames in order to have video level feature vector that represents the video. We applied some dimensionality reduction techniques before this step, and then trained classifiers like SVM and Logistic regression on the video feature vector. The obtained accuracy was around 75%. The second method is using pretrained VGG16 network, which was trained to classify 1000 objects. We use it to extract the features for each frame from the second last layer to construct frame level features, then we construct video level features in a similar way as mentioned before. This is followed by training classifier like MLP, SVM, and Logistic regression to classify the videos and it gave accuracy of approximately 90% which proves that using bigger pool of objects increased the accuracy significantly. We proved that using few key frames as representation of a video can give enough information about the video. In addition, we can use objects to classify scenes successfully, and the more object detectors we have, the higher the accuracy will be. Moreover, using dimensionality reduction techniques helps a lot with processing features for videos, in regards to running time and memory. Finally, we learned that there are many existing online pre trained models like VGG16, which can be used directly for transfer learning.

7 Future Work

In the future ,we are planing to expand this project in variety of ways. Firstly, we are planning to grow our dataset by having at least 100 videos for each sports categories. Secondly, we are going to try more sophisticated video summarization techniques like VSUMM, LSTM, etc. to extract keyframes as compared to the naive approach of using intensity difference across consecutive frames. Since this step is really crucial in transforming videos to a set of frames, an advancement in this part should give us further boost to accuracy. Instead of just using the keyframes, we can use RNN to process all the frames of the video, and hence get a better representation for the whole video. As more and more pre trained models are getting available, we will work on experimenting with different pre-trained models such as GoogleNet, VGG19, and AlexNet e.t.c on our dataset. Using other pre-trained models can give us more information about how to extract features from pre trained network. Finally, we aim to transform this application to a real-time video classification which can be run on a browser. We believe that there is no limit for extending and enhancing this project, as the field of machine learning is advancing at a very fast pace and the utility of classifying video scenes is enormous in various walks of life.

All the code for this project can be found on : https://github.com/amanwalia92/SportsVideoClassification

References

[1] Li, Li-Jia and Su, Hao and Lim, Yongwhan and Fei-Fei, Li, *Objects as Attributes for Scene Classification*, Computer Science Department, Stanford University

[2] Li-Jia Li and Li Fei-Fei. *What, where and who? Classifying event by scene and object recognition* IEEE Intern. Conf. in Computer Vision (ICCV). 2007

[3] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar and Li Fei-Fei. *Large-scale Video Classification with Convolutional Neural Networks* CVPR, 2014

[4] D. Lowe. *Object recognition from local scale-invariant features.* In Proc. International Conference on Computer Vision, 1999.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei.*ImageNet: A Large- Scale Hierarchical Image Database.* In CVPR09, 2009.

[6] Leonardblier, Par. *A Brief Report of the Heuritech Deep Learning Meetup #5.* Blog.heuritech.com, 29 Feb. 2016, blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/

[7] Guyon, Isabelle & Elisseeff, André. (2003) *An Introduction of Variable and Feature Selection. J. Machine Learning Research Special Issue on Variable and Feature Selection.*

[8] Zha, Shengxin , Luisier, Florian , Andrews, Walter , Srivastava, Nitish , Salakhutdinov, and Ruslan. (2015). *Exploiting Image-trained CNN Architectures for Unconstrained Video Classification.* 10.5244/C.29.60.