Recognition of Normal Action and Criminal Action of a Human based on Skeletal Key-point Technique

Rohan Bhatkande
School of Computing
Dublin City University
Dublin, Ireland
rohan.bhatkande2@mail.dcu.ie
20210678

***Abstract*—Recognising a human action is one of the complex tasks a computer can perform. More often building a model that differentiates between two distinct actions of a human being is quite easy to implement but to build a model that differentiates multiple similar, as well as distinct actions of a human being, is quite challenging. So to overcome this problem we have built an action recognition model which recognises human action using the human skeletal key-point detection technique. To perform real-time multi-person skeletal key-point detection we have used the OpenPose library which detects 135 key-points in total. To perform the action recognition we have implemented AdaBoost and MLP algorithm. We have performed a two-class and multi-class classification using imbalanced data and balanced data in which MLP with balanced data achieved the highest f1-score in the two-class classification and in the multi-class classification scenario. In this experiment, we examine the behaviour of a model when a complicated real-time situation arises and how it impacts the performance of the model as seen between two-class and multi-class classification. The UCF-crime dataset was used to build this model which was made up of live video footage obtained from the surveillance cameras.**

***Index Terms*—UCF-crime, Multilayer Perceptron(MLP), Adaptive Boosting (AdaBoost), OpenPose, Skeletal Key-point Detection, Synthetic Minority Oversampling Technique(SMOTE),Principal Component Analysis(PCA), Action Recognition.**

I. Introduction

We as humans have become so advanced in the field of computer vision that we have started using it to map, track, and recognise the things that happen around us. One such instance is the behaviour of a human being, how a person behaves in public places is one of the most critical factor that highlights the safety of oneself. Even though there are surveillance cameras installed at every point around the corner in cities, for human safety and to detect suspicious activity, it is quite difficult to track and check every human action manually. The capability of a human being to detect a crime on the monitor decreases as the number of screens increases. Also to watch a live video for long hours is difficult for a human being as there will be disturbance and distraction which may lead to scattering of attention [1]. To overcome this situation we can recognise human actions automatically without any interruption by using machine learning. In this paper, the skeletal key-point approach is implemented, where the key-points of a human body coordinates are used to determine the human action, which is one such study that we will be focusing on to build a robust action recognition model. To perform this experiment a publicly available dataset called UCF-crime was used which was obtained from surveillance cameras or security cameras. Further Openpose was used to extract the body key-points from the frame(image) and feed the preprocessed data to an ensemble learning method and artificial neural network. While building an action recognition model there was one such factor that was emphasized to create a more optimized model which was a balanced dataset and also to understand how balanced data helped to improve the performance of the model. Moreover, quite often an image with a single-person action having a static background is used to build an action recognition model, whereas in this paper we have used the video data where multiple people are present in a dynamic background while performing an action. Also, another focus of our study will be how the different machine learning models behave for two-class and multi-class classification which will be explained further.

The rest of the paper is built as section 2 which is related work that consists of a literature review that was carried out, section 3 is based on the methodology which highlights the techniques used and the proposed approach, section 4 is based on results and discussion about the work and section 5 is concerned with the conclusion and future work.

II. Related Work

To build an action recognition model the initial requirement is the type of data to be taken into consideration which can further be classified as the dataset that is based on static background or dynamic background [2]. Human activity is categorized in different manners which are based on events, gestures, atomic actions, behaviour, group actions, human-to-human and human-to-object interaction [3]. For our experiment, we are using the UCF-crime dataset, which has a dynamic background, which is a collection of 8 different human acts namely arrest, arson, abuse, assault, burglary, explosion, fighting and normal action [4].

Although there are many ways to build an action recognition model using various techniques such as image segmentation [5], boundary detection [6] and pose estimation [7]. In this paper, human pose estimation is one such approach that we will be implementing which helps to locate the key-points of a human body that further helps to build a more powerful action recognition model. Gupta, Abhay et al. [8] proposed an approach where a human action recognition and classification can be determined by its skeletal pose. The data that they have focused on is of a still image with static background, although the results obtained after implementing an action recognition is quite acceptable but the amount of data which is just 1000 images of 5 different human activity could be questionable.

J. Talukdar, B. Mehta et al. [9]proposed an approach which combines the good features with an optical flow algorithm to get the feature vector which is later classified using Multilayer Perceptron(MLP). There are two main aspects that this paper talks about to improve the accuracy of a machine learning model which are by increasing the training sample size and also the number of hidden layers in Multilayer Perceptron(MLP). Although the accuracy obtained by their model is promising yet the number of the hidden layer(ie. 200) is something that would be examined. In general, increasing the number of hidden layers does improve accuracy but it also impacts the time complexity of the model [10]. Also, the feature vector size played an important factor while achieving better accuracy, according to the author.

According to the author [11], to build an action recognition model image segmentation was implemented on the 5000 images(static and dynamic background) of 6 posture classes and studied the nature of various algorithms like K-nearest neighbour, Support Vector Machine, Naive Bayes, Neural Network and AdaBoost. Although AdaBoost gave a decent result of 96.56% on imbalanced classes. However, if the classes would have been balanced the output would vary.

There are many ways to improve the performance of a deep learning model, one such is by sampling the data according to the author [12]. There are two different types of datasets namely trimmed(relevant clips with action) and untrimmed(mixture of relevant and irrelevant clips with action). The untrimmed dataset is based on real-life situations. Although there are 5 different datasets that the author has worked on. One amongst them is the UCF-crime, untrimmed dataset, which is of our interest too. The experiment was based on temporal motion information and action recognition where 4 different architectures were implemented to detect if it is a criminal or normal action. The maximum accuracy obtained was 82.12% for GCN+TSN-RGB[10] architecture on the UCF-crime dataset.

III. Methodology

In this section, we will describe the methods, techniques and dataset that were used to perform this experiment. This experiment has two main methods namely skeletal key-point detection and action recognition. Figure 1 shows the pipeline of the system.

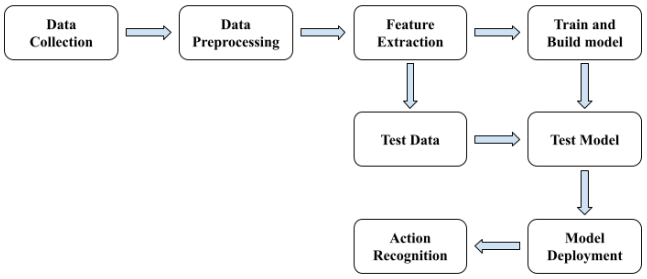

Fig. 1. Proposed Architecture

A. Data Collection

Collecting the required dataset plays an important role while building a machine learning model. The data used in this experiment is in video format. The frames are further extracted from the videos at the rate of 30 fps and are stored in their respective folders with a resolution of 320*240 pixels. Figure 2 shows the extracted frames from the videos of different classes.

Fig. 2. Extracted Frames

B. Data Preprocessing

The extracted frames are then fed to a human pose estimation framework known as OpenPose to get the skeletal key-points of a human body.

1) OpenPose: OpenPose is a real-time multi-person system that detects the human body, foot, and facial key-points and uses Part Affinity Fields to learn the body parts along with the individuals in the image [13]. One advantage of OpenPose is that it is a bottom-up system which achieves higher performance and efficiency irrespective of the number of people in the image. Figure 3 represents the pipeline of OpenPose. The working of OpenPose is that it takes input as an entire image to the Convolutional Neural Network(CNN). CNN analyses the entire image and provides a set of feature maps which is provided as input to the first stage.

In the first stage, confidence maps are a 2D representation of a given pixel where particular body parts can be located. In general, body parts are detected by the confidence maps.

Although the part association is achieved by Part Affinity Fields it does so by preserving the location and orientation information across the region of support of the limb. The bipartite matching associates body part candidates. In the bipartite graph, the nodes are body part detected candidates and the edges are connections between the pairs of detected candidates. In a bipartite graph, edges are selected in a manner where no two edges share the same node. Parsing results assemble them into full body poses for all the people in the image [13]. The pipeline of OpenPose is shown in figure 3.

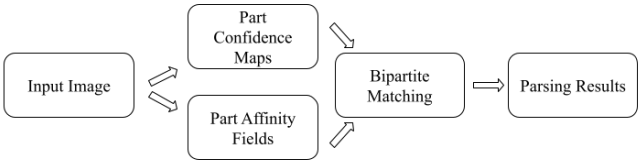

Fig. 3. Openpose Pipeline

After feeding the input images to OpenPose the output images obtained have key-points structure on a human body which is shown in figure 4.

Fig. 4. openpose processed images

C. Feature Extraction

Once the skeletal key-points for each of the images are obtained. The skeletal data will be used as the feature to obtain the body velocity, joint velocity and normalized joint. The processed data is further classified as features and labels. The images which do not exhibit key-points will be removed in our experiment. Further, the data is divided into training and testing data with a ratio of 70:30 respectively. Also to deal with the imbalanced data we implement an approach called SMOTE and to reduce the dimensionality of the input data we implement Principal Component Analysis.

1) Over Sampling: To deal with the class imbalance which is the unequal distribution of classes that usually arises in classification we implement a method called resampling. There are two types of resampling namely oversampling which balance the class spread by random repetition of minority class samples and undersampling which balances the class spread by eliminating samples randomly from the majority class. In this experiment, we will be focusing on the over-sampling technique named Synthetic Minority Oversampling Technique(SMOTE). The main idea of oversampling is that it increases the number of minority classes which balances the class spreading through minority target instances [14].

2) Principal Component Analysis: Principal Component Analysis(PCA) is a dimensional reduction technique that helps to reduce the size of the data without the loss of important information. It not only makes analyzing of data easier but also reduces the requirement of higher computational cost [15]. In this experiment, PCA is implemented on the training data to reduce the dimensionality. For example, in this experiment we have got variance of 94% on balanced training data which means that we have preserved 94% of the information of the original data after applying PCA.

D. Train and Build Model

The training data is fed as an input to AdaBoost and Multi-layer Perceptron respectively.

1) AdaBoost: Adaptive Boosting(AdaBoost) is a type of boosting technique which combines the prediction of multiple weaker classifiers. Adaboost assigns weights to each training sample and miss-classified samples are assigned with higher weights so that they are visible as training subsets with a higher probability for the next classifier. One advantage of AdaBoost is that it mainly focuses on the samples which are missclassified. It is not concerned with the issue of overfitting. However, there is one disadvantage of this approach that it is sensitive to noise data [16].

2) Multi-Layer Perceptron: Multilayer perceptron(MLP) is a feed-forward neural network which is a type of Artificial Neural Network. It is consist of 3 layers namely the input layer which accepts the input to be processed, the hidden layer which acts as a computational unit and the output layer which performs the task of classification. One advantage of MLP is that it learns non-linear models. However, the disadvantage of MLP is that it is sensitive to feature scaling. There is no defined technique to determine the number of neurons in the hidden layer thus it can only be achieved by performing trial and error [17]. Figure 5 represents the architecture of MLP which is referred from this study [18].

This architecture shows the connection between the layers along with their assigned weights. The surrounding provides values to the input layer, and the values of all other neurons are calculated by the weights and values of previous layers. To elucidate, suppose we have a hidden layer (h5) which has input from previous hidden layer (h1) and hidden layer (h2) along with their weights w8 and w9. We can formulate the h5 node as,

$$h5 = h1.w8 + h2.w9$$

Values from the previous layers are transformed in the hidden layer which is followed by a non-linear activation function such as ReLU. Furthermore, the values obtained from the

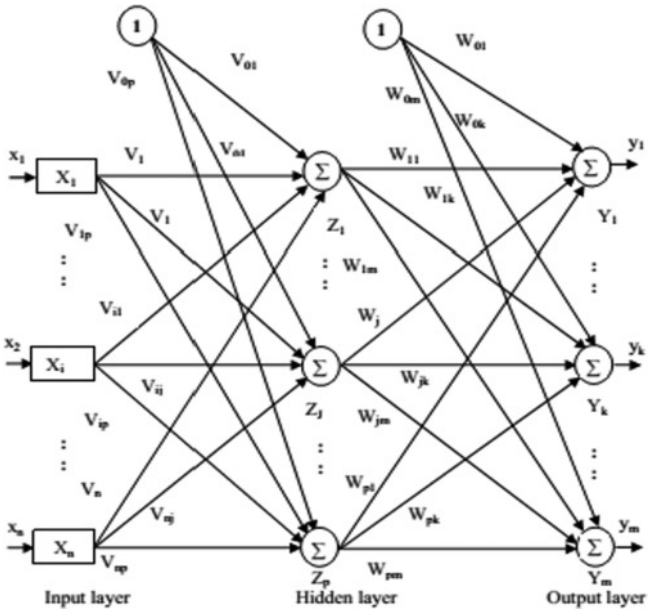

Fig. 5. Architecture of Multi-Layer Perceptron

hidden layers are transformed into output values at the output layer. Also in the case of multi-class classification softmax is used as the output function which calculates the probability of each class [19] or else for binary classification the activation function used is always sigmoid but in this experiment since we are using scikit-learn MLPClassifier [20] we will be using ReLU as our activation function. Adam is the optimizer that is being used in this experiment. The weights are assigned randomly before the start of training and after completion of the training set, the model gets validated. The output of the network is dependent on the neuron and the weight of the network.

3) ReLU: Most of the time many users prefer to use softmax as an activation function in the artificial neural networks without realising its drawback. Also when sigmoid and tangent hyperbolic functions are used, as the number of layers increases it gives rise to vanishing gradient problem [21]. Thus to overcome gradient decent problem Boltzmann Machine, Nair et al. [22]introduced the rectified linear units. ReLU is a non-linear activation function which behaves in a linear manner. The mathematical form is as shown below which implies that the function is linear for values greater than zero. In general terms, ReLU will always be either 0 or greater than or equal to 1.

$$g(z) = \max(0, z)$$

where z represents a numerical value

E. Test Data and Test Model

The test data obtained in the feature extraction phase is used to test the accuracy of the model. We plot the confusion matrix to understand the actual vs the predicted values that were generated by the model. We use the performance metrics to understand the working of our system. Also higher the F1-score better the model performs.

F. Model Deployment

Once the model is tested it can be deployed on various platforms like live surveillance cameras or checked directly on a pre-recorded video or image. Although the model performs pretty well on live surveillance cameras however it gives promising results on the pre-recorded videos or images.

G. Action Recognition

There are three different methods by which action recognition can be implemented namely the Body-Based model where action recognition is based on 2-Dimensional(2D) and 3-Dimensional(3D) features like skeletal key-points and segmentation. Bag of visual words model which uses a bag of visual words and local features. The local features are extracted by using a point detector and also by extracting local descriptors. These descriptors are later clustered into visual words. Deep Learning Approach is another method that performs action recognition which uses models like CNN to perform classification using images or videos [23].

In this experiment, we are using the Body-Based model and Deep Learning Approach to build an action recognition model that recognises the actions of the people in a video. There are many factors that impact an action recognition model. These factors include the visibility of a person, the saturation of the image, brightness, and distance of a person from the camera. Once all the requirements are matched, the action recognition provides the relevant action of the person in an image or video.

H. Evaluation metrics for Classification

1) Confusion Matrix: When dealing with a classification-based problem, the performance of the model is evaluated by building and understanding a confusion matrix which has True Positive(TP) which means that the positive class is predicted correctly as positive, False Positive(FP) which means that the negative class is predicted incorrectly as positive, True Negative(TN) where that the negative class is predicted correctly as negative and False Negative(FN) which means that the positive class is predicted incorrectly as negative. A confusion matrix is a representation of actual values and predicted values [24].

2) Precision or Positive Predicted values: Precision is a measure of positively correct prediction. Precision is calculated as true positive(TP) divided by the sum of true positive(TP) and false positive(FP). The formula of Precision is as shown below.

$$Precision = \frac{TP}{TP + FP}$$

3) Recall or sensitivity: Recall is measure of positively correct prediction out of all the positive predictions that could have been built [25]. The recall is calculated as true

positive(TP) divided by the sum of true positive(TP) and false negative(FN). The formula of Recall is shown below.

$$Recall = \frac{TP}{TP + FN}$$

4) F1-score(macro averaging): F1-score is a measure that combines precision and recall and provides their output as one. Mathematically, it is the harmonic mean(HM) of precision and recall. There are two main averaging techniques namely micro and macro. However, to deal with imbalance instances it is always advisable to go ahead with macro averaging as it treats all the classes equally without taking the amount of data availability into consideration [20]. The mathematical form of the f1-score is shown below.

$$F_1 = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5) Accuracy: Accuracy is a metric that is used to define the performance of a classification system. It is a measure of the ratio of true positive(TP) and true negative(TN) divided by all the possibilities of positive and negative observations. However, during the instances of Imbalanced class, accuracy is not a relevant metric to look for. The mathematical form of the accuracy is represented below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

I. Dataset

The dataset used was a subset of UCF-crime, built from the main UCF-crime dataset [4], which consists of 8 different actions namely abuse, arrest, arson, assault, burglary, explosion, fighting and normal behaviour. Each of the action folders consists of video files ranging from 50 to 100 respectively. So to perform this experiment we have used all the actions specified except explosion as we are detecting human action based on skeletal key-point. A total of 50 videos have been taken to perform this experiment. Frames are extracted from these videos at the rate of 30 frames per second. The frames or the images that were extracted were in the range of 1,15,000 to 2,94,000 per action respectively. However, from the frames collected only the frames or the images with visible human action had to be selected.

For binary classification, images were selected with visible human beings performing an action where a total of 18769 out of which 17956 images had detected skeletal key-points which was further used to train the model. After extracting the time serials feature the total training data was a total of 15757 samples. The classes used to perform binary classification were normal action which had 10156 samples and assault action which had 5601 samples.

Similarly for multi-class classification, a total of 50,622 images were used with visible human action out of which 41,409 images had detected skeletal key-points. The classes used to perform multi-class classification were normal action which had 10156 samples, assault had 5601 samples, abuse had 1184 samples, arrest had 3718 samples, arson had 6316 samples, burglary had 7968 samples and fighting had 6466 samples respectively. An overview of the class distribution can be see in table 1.

IV. Results and Discussion

This experiment majorly focuses on how AdaBoost and Multi-Layer Perceptron behaves in a two-class and multi-class classification environment. Also, we have focused on how to improve the performance of the model by balancing the dataset which will be explained further.

A. For Binary/Two-Class Classification

Binary Classification was performed using imbalanced and balanced training data and the summary of accuracy achieved is shown in Table 2.

1) Imbalanced Data: A total of 11029 training samples and 4728 testing samples were used to perform this experiment.

The normal class had 7126 training samples and assault had 3903 training samples. AdaBoost had a training accuracy of 83% and a testing accuracy of 82%. Also, the accuracy of the normal class was recorded as 89% which was quite higher compared to the assault class which recorded an accuracy of 70%. To improve the performance of the model we implemented another algorithm namely MultiLayer Perceptron(MLP) which accounted for training accuracy of 99% and testing accuracy of 90% along with the normal class exhibiting an accuracy of 93% and assault class as 85%.

Although we had achieved promising results, however, to improve the performance of the model we initiated an approach to build a balanced training dataset and study how it impacts the performance of Adaboost and MLP.

2) Balanced Data: A total of 14252 training samples and 4728 testing samples were used to perform this experiment.

After applying SMOTE technique, the minority class samples were increased to that of the majority class such that both the normal class and the assault class have 7126 samples each for training. AdaBoost achieved a training accuracy of 81% and testing accuracy of 81%. Also, the normal class accuracy was 82% and assault class was of 80%. Whereas, MLP on the other obtained a training accuracy of 99% and testing accuracy of 91%. The accuracy per class was 93% and 86% for normal and assault class.

After implementing SMOTE and balancing the training data the accuracy of AdaBoost decreased by 1% when compared to the imbalanced class. Although the accuracy of the AdaBoost is not up to mark one thing to notice in this experiment is that there is no huge gap between the training and testing accuracy which means that the AdaBoost is robust to overfitting which is not the case in MLP. But when SMOTE was implemented on MLP the accuracy of the model increased to 91%, which was 1% increase, from 90% which was achieved by the imbalanced data. Although the accuracy obtained is quite promising, since we had imbalanced data, we have to substitute Precision, Recall and F1-score. The precision has been decreased when the model was trained on balanced data rather than imbalanced data whereas recall on the other hand increased in the balanced

TABLE I
CLASS DISTRIBUTION

Class	Number of Samples
Normal	10156
Assault	5601
Abuse	1184
Arrest	3718
Arson	6316
Burglary	7968
Fighting	6466

data scenario which could be directly linked to the increase in the amount of data while training the model. Another main point to notice in binary classification was the F1-score achieved by AdaBoost and MLP in imbalanced and balanced data experiment was nearly consistent with a slight change in decimals as shown in Table 4.

B. For Multi-Class Classification

Multi-Class Classification was performed using imbalanced and balance training data and the summary of accuracy achieved is shown in table 3.

1) Imbalanced Data: A total of 28986 training samples and 12423 testing samples were used to perform this experiment. A total of 7 different classes were used to build a multi-class classification model. The sample of training data in each class varied such that normal class had 7123 samples, assault had 3903 samples, abuse had 812 samples, arrest had 2610 samples, arson had 4412 samples, burglary had 5580 samples and fighting had 4546 samples.

When AdaBoost was implemented on the given data the training accuracy obtained was 45% and testing accuracy was 44%. where each class had achieved accuracy ranging from 15% up to 79%.

Also, when MLP was implemented the training accuracy achieved was 87% and testing accuracy was 77%. The accuracy range for each class range from 62% to 91% as shown in figure 7. The class with the highest accuracy was normal class and the one with the lowest accuracy was assault and abuse class for both the models.

2) Balanced Data: A total of 49861 training samples and 12423 testing samples were used to perform this experiment. The training data has a consistent number of samples which was 7123 throughout the different classes.

In AdaBoost, the accuracy obtained for training was 41% and testing was 40%.The accuracy obtained by each class range from 24% to 64%.In MLP, the accuracy of training data was 90% and test data was 78%. The accuracy for each class ranged from 65% to 90% and the class that obtained higher accuracy was normal class and the lowest was assault class in AdaBoost.

In MLP, the class that obtained the highest accuracy was arrest class and the class with the lowest accuracy was assault and fighting class. Since we had imbalanced class we had to calculate the precision, recall and F1-score of the model to reach the conclusion. The F1-score got decreased from 37.67% to 36.53% in AdaBoost for balanced trained data. But in MLP, the F1 score increased from 75.25% to 77.15% when implementing balanced data multi-class classification as shown in table 4.

From the results obtained in multi-class classification for balanced and imbalanced data, we can see that after implementing SMOTE technique on training data there was increase in performance of the MLP model when compared to the imbalanced trained data. However, for balanced trained data the accuracy significantly decreased in AdaBoost.

From the results obtained we can clearly examine the behaviour of Adaboost and MLP during binary and multi-class classification for imbalanced and balanced trained data. In binary classification, MLP performs better compared to AdaBoost but when multi-class classification has been implemented the accuracy of the MLP decreases significantly compared to two-class classification which is directly linked to the increase in complexity of the multi-class classification model. Also, assault class saw a significant decrease in accuracy in the multi-class scenario when compared to binary classification and the reason for it can be the increase in action classes. Even if we have a sufficient amount of data we cannot achieve higher accuracy for the similar action-based class as the model would often confuse itself as seen in the assault, burglary and fighting class. The confusion matrix is obtained by comparing the ground truth with the predicted class and it is represented as shown in figure 6 which highlights the classes that were recognised correctly and the classes recognised falsely.

Although the model performs relatively well in recognising human actions at the same time there are certain factors that need to be taken into consideration such as the angle of the camera, which is important to fetch the human body skeletal key-points to understand the type of action the human is actually performing and to deal with this kind of situation multiple cameras at different angles can be used. Also, the distance of a person from the camera also impacts the recognition system behaviour so a more close view can be a solution for the given situation. The output of the model has been shown in figure 7 and figure 8.

V. CONCLUSIONS AND RECOMMENDATIONS

In this paper, we have performed two-class and multi-class classification with the balanced data approach along with the OpenPose framework to extract the skeletal key-points from an image. A comparative analysis has been performed

TABLE II
RESULTS SUMMARY OF BINARY CLASSIFICATION

Binary Classification							
Trained on ImBalanced data				Trained on Balanced data			
AdaBoost		MLP		AdaBoost		MLP	
Class	Accuracy	Class	Accuracy	Class	Accuracy	Class	Accuracy
Normal	89%	Normal	93%	Normal	82%	Normal	93%
Assault	70%	Assault	85%	Assault	80%	Assault	86%

TABLE III
RESULTS SUMMARY OF MULTI-CLASS CLASSIFICATION

Multi-Class Classification							
Trained on ImBalanced data				Trained on Balanced data			
AdaBoost		MLP		AdaBoost		MLP	
Class	Accuracy	Class	Accuracy	Class	Accuracy	Class	Accuracy
Normal	79%	Normal	91%	Normal	64%	Normal	89%
Assault	15%	Assault	62%	Assault	24%	Assault	65%
Abuse	22%	Abuse	62%	Abuse	46%	Abuse	74%
Arrest	34%	Arrest	87%	Arrest	55%	Arrest	90%
Arson	36%	Arson	80%	Arson	30%	Arson	82%
Burglary	51%	Burglary	77%	Burglary	34%	Burglary	75%
Fighting	25%	Fighting	64%	Fighting	25%	Fighting	65%

TABLE IV
PERFORMANCE EVALUATION

Results				
Algorithms	Accuracy	Precision	Recall	F1-measure
ImBalanced-AdaBoost(Binary)	82.00%	81.40%	79.73%	80.41%
Balanced-AdaBoost(Binary)	81.00%	80.00%	81.28%	80.48%
ImBalanced-MLP(Binary)	90.00%	89.74%	89.24%	89.48%
Balanced-MLP(Binary)	**91.00%**	**89.61%**	**89.38%**	**89.49%**
ImBalanced-AdaBoost(Multi)	44.00%	40.54%	37.60%	37.67%
Balanced-AdaBoost(Multi)	40.00%	36.22%	40.00%	36.53%
ImBalanced-MLP(Multi)	77.00%	75.81%	74.82%	75.25%
Balanced-MLP(Multi)	**78.00%**	**76.85%**	**77.51%**	**77.15%**

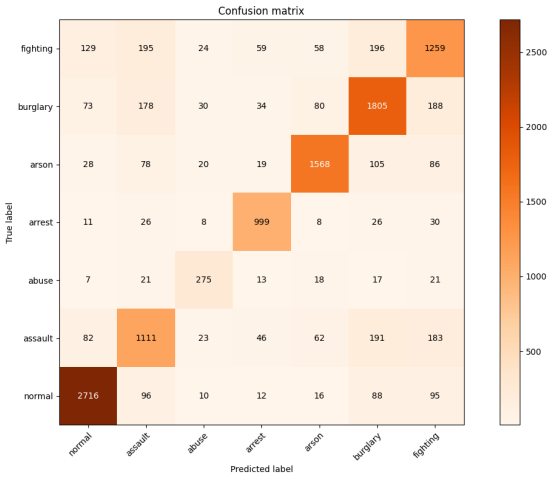

Fig. 6. Confusion Matrix of MLP trained on balanced training data

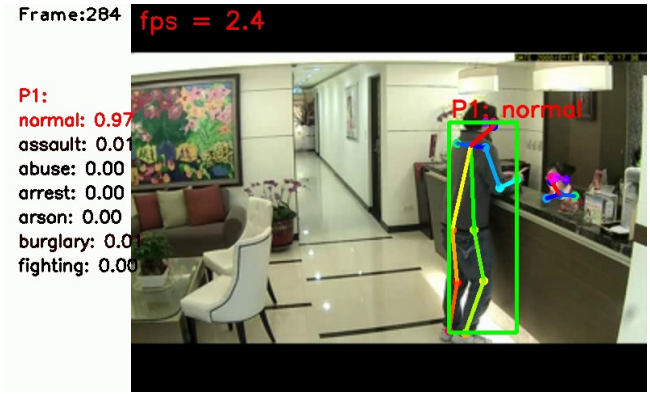

Fig. 7. Multiclass recognition output with the entire human body and closer camera view

between different classifiers. In two-class classification, MLP with balanced data outperformed AdaBoost with balanced and imbalanced data and MLP with imbalanced data. In multi-class classification, MLP with balanced data outperformed AdaBoost with balanced and imbalanced data and MLP with imbalanced data. As we are more concerned with the real-time situation we are more interested in multi-class classification rather than binary classification since it provides a complex task for the model to work on. It was extremely difficult to perform this experiment as it was based on a real-world crime video dataset with a dynamic background, unlike the

Fig. 8. Multiclass recognition output with the entire human body and but not closer camera view

past experiments which were based on the still image with a static background.

A. Future Work

Future work for this experiment could be to use Spatio-temporal feature learning network for human action recognition. Also to improve the accuracy of the model we can implement post-processing techniques. Although our approach only detects the human body further experiment could be done to recognise an animal or child to classify the actions such as animal abuse and child abuse. We can also implement different deep learning models with changes in hyper-parameter to understand their impact on the performance.

VI. Acknowledgement

We would like to thank our supervisor, Dr Suzanne Little, for her consistent support and guidance during the tenure of this project.

References

[1] G. van Voorthuijsen, H. van Hoof, M. Klima, K. Roubik, M. Bernas, and P. Pata, "Cctv effectiveness study," in *Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology*, 2005, pp. 105–108.

[2] S. Kang and R. P. Wildes, "Review of action recognition and detection methods," *CoRR*, vol. abs/1610.06906, 2016. [Online]. Available: http://arxiv.org/abs/1610.06906

[3] K. I. A. Vrigkas Michalis, Nikou Christophoros, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, 2015. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2015.00028

[4] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," *CoRR*, vol. abs/1801.04264, 2018.

[5] S. Arseneau and J. Cooperstock, "Real-time image segmentation for action recognition," in *1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1999). Conference Proceedings (Cat. No.99CH36368)*, 1999, pp. 86–89.

[6] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," *CoRR*, vol. abs/1705.01180, 2017.

[7] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang, "The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation," *IEEE Access*, vol. 8, pp. 133 330–133 348, 2020.

[8] A. Gupta, K. Gupta, K. N. M. Gupta, and K. O. Gupta, "Human activity recognition using pose estimation and machine learning algorithm," in *ISIC*, 2021.

[9] J. Talukdar and B. Mehta, "Human action recognition system using good features and multilayer perceptron network," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, 2017, pp. 0317–0323.

[10] M. Uzair and N. Jamil, "Effects of hidden layers on the efficiency of neural networks," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1–6.

[11] N. Zerrouki, F. Harrou, Y. Sun, and A. Houacine, "Adaboost-based algorithm for human action recognition," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 189–193.

[12] A. F. D. Marsiano, I. Soesanti, and I. Ardiyanto, "Deep learning-based anomaly detection on surveillance videos: Recent advances," in *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 2019, pp. 1–6.

[13] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," vol. 43, no. 1, 2021. [Online]. Available: https://doi.org/10.1109/TPAMI.2019.2929257

[14] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, "Smote for handling imbalanced data problem : A review," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1–8.

[15] A. Agarwalla, D. Dileep, P. Jyothsana, P. Unnikrishnan, and K. Thirumala, "Principal component analysis and cnn for classification of power quality disturbances," in *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2021, pp. 1–5.

[16] Y. Zhang, M. Ni, C. Zhang, S. Liang, S. Fang, R. Li, and Z. Tan, "Research and application of adaboost algorithm based on svm," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, pp. 662–666.

[17] U. Orhan, M. Hekim, and M. Ozer, "Short communication: Eeg signals classification using the k-means clustering and a multilayer perceptron neural network model," vol. 38, no. 10, 2011. [Online]. Available: https://doi.org/10.1016/j.eswa.2011.04.149

[18] M. D. Mohanty and M. N. Mohanty, "Chapter 5 - verbal sentiment analysis and detection using recurrent neural network," in *Advanced Data Mining Tools and Methods for Social Computing*, ser. Hybrid Computational Intelligence for Pattern Analysis, S. De, S. Dey, S. Bhattacharyya, and S. Bhatia, Eds. Academic Press, 2022, pp. 85–106. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323857086000126

[19] Y. Lu and S. Shetty, "Multi-class malware classification using deep residual network with non-softmax classifier," in *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, 2021, pp. 201–207.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] H. Ide and T. Kurita, "Improvement of learning for cnn with relu activation by sparse regularization," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2684–2691.

[22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines." Madison, WI, USA: Omnipress, 2010.

[23] M. Burić, M. Pobar, and M. I. Kos, "An overview of action recognition in videos," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 1098–1103.

[24] V. A. Narayana, A. Govardhan, and P. Premchand, "To create a confusion matrix in respect of threshold being fixed for effective detection of near duplicate web documents in web crawling," in *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, 2011, pp. 763–768.

[25] A. Bansal and A. Singhrova, "Performance analysis of supervised machine learning algorithms for diabetes and breast cancer dataset," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 137–143.