Vingroup Big Data Institute

BML - Project Report

Helmet Impact Detection

Hieu Tran | Khanh Pham | Hieu Luu

December 27, 2020

Abstract

This project is for the Basic Machine Learning course. The topic of the project is to solve the helmet impact detection competition posted on Kaggle by the National Football League(NFL). America football is considered the most popular sport in America with both viewership and personal preference. However, the sport has been questioned in the last few years for it safety, especially with regard to head trauma and chronic traumatic encephalopathy (CTE). This scale and difficulty of the project was chosen to in order to challenge the knowledge of the group as well as to force its members to learn and explore more in the topic of computer vision in general and object detection specifically. The dataset provided by the problem consisted of videos of game plays, including 2 views: end view and sideline. Furthermore, player tracking data are also provided, which is composed of: the coordinate of the player on the field, the player's velocity, acceleration, orientation and angle of motions. Due to the short time frame of the project, the data used mostly for solving the task of impact detection are the video data. The tracking data was explored but no complex solution was developed with the tracking data in used. The task was split into two smaller, more manageable tasks: helmet detection and impact classification. The project was able to achieve good result with the helmet detection class using a Faster R-CNN model. Using the predictions of the helmet detection model, different approaches were tried for the classifying model with mixed result. Overall, this project can be developed further to achieve better result, especially diving deeper into the tracking data as well as studying the will be published approaches of successful teams in the competition.

Contents

1 Background

American football is considered the most popular sport in America in term of viewership and general public interest. In a survey by Gallup in 2018, 37% of American answered that football is their favourite sport and in 2020, the Super Bowl achieved nearly 100 million viewers. However, growing concern about the sport safety has been brought up over recent years, especially with regard to brain trauma and chronic traumatic encephalopathy (CTE), a degenerative brain disorder related to repeated head trauma. A recent study found that 87% of 177 players across all levels and 99% of 111 former professional players exhibited symptoms of CTE. The health consequences of head trauma and CTE can range from life discomfort and hardship to severe such as dementia and other cognitive dis-functions.

This project's aim is to solve the challenge created by the NFL to detect helmet impact from video posted on Kaggle. This competition is part of the NFL's annual 1st and Future competition, which is designed to spur innovation in athlete safety and performance. If successful, this project could provide significant support to the effort of reducing the head trauma in America professional football the research program of the NFL.

The helmet impact detection problem was break down into two sub-problems: an object detection task to identify the players' helmets from images and a binary classification task to determine whether or not the helmet is in a collision in the frame of the video. The helmet detection problem is an image object detection problem as the number of helmets appearing in a frame of the video may change significantly from frame to frame.

As this is a project submitted for the course of Basic Machine Learning, the project not only tested different deep learning computer vision approaches but also classical machine learning algorithms. The group is consist of three computer science fresh graduates who have taken at least one machine learning course or artificial intelligence course at university. In order to acquire more knowledge and experience in computer vision, a particularly difficult task was chosen. Any attempt at solving the problem will require a combination of knowledge provided from the course as well as prior knowledge of the member in the field of computer vision.

2 Literature Review:

2.1 Artficial Neural Networks:

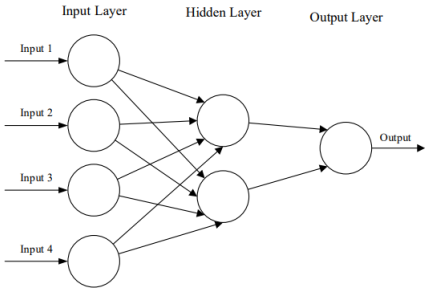

Figure 1: Structure of ANNs

Artificial neural networks (ANNs) are computational processing systems inspired by the biological structure and mechanic of biological nervous systems (biological brains)[1]. ANNs are often consisted of multiple connected layers of computational nodes called neurons, which represent the biological neurons. These neurons have a set of input values, a set of associated weights, a bias value and an activation function. When a neuron reads an input set coming in, it calculates the weighted sum of the values in the input set, then add the value with the bias term before passing the result through its activation function. Without the activation function, a layer of neurons will perform a linear transformation on the input values. The non-linear activation functions allow ANNs to approximate the output of all functions linear or non-linear given sufficient complexity and amount of data points for the network to learn from.
ANNs are composed of three sets of neuron layers: input, hidden and output. An artificial neural network is considered a deep neural network when the number of neuron layers stacked upon each-other within the hidden layer is more than two. In recent years, deep learning has been applied successfully into many applications such as computer vision, speech recognition, natural language processing, etc.

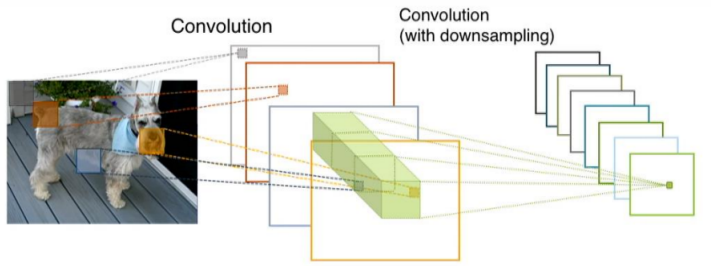

Figure 2: Example of CNNs extracting features from image

2.2 Convolutional Neural Network:

One of the weakness of traditional ANNs is the struggle with the processing and analysis of image data. With traditional ANNs, when passing an image through the network, the image is required to be flatten out to a vector. For example, in the well-known MNIST dataset to classify hand-written digit images with each image is a black and white 28 pixels by 28 pixels image, the images are convert into a vector of length $1*28*28 = 784$. The number of input values for each node in the input layer will have to match the input size of the vector. However, when considering image with much higher resolution and 3 dimensions for a RGB image, the number of parameters included in the network will sky rocket. Furthermore, when an image is flatten out, the spatial information and structure contain in the image is discarded. These information can be detrimental to the accuracy of the ANNs when performing the tasks required.

The fully connected layers in traditional ANNs are replaced with 2D kernels that is used to extract features from the input image. Each kernel has the same depth as the image. When processing an image with a kernel, the kernel is slid over the image. The dot product of the values within the kernel and the image pixels is calculated to be put in the center point of the current position of the kernel on the activation map. Using these kernels helps CNNs reduce the number of parameters that required to be tuned comparing to ANNs. Different parameters of the convolution such as stride and padding can affect the size of the activation map. As convolution is a linear transformation, an activation is needed to make stacking convolution layers upon each other effective in learning the problem. The activation map after passing through the different convolution layer of the CNNs, will often be flatten as pass through one or more fully connected layer in order to achieve the wanted

result.

2.3 Image classification and object detection

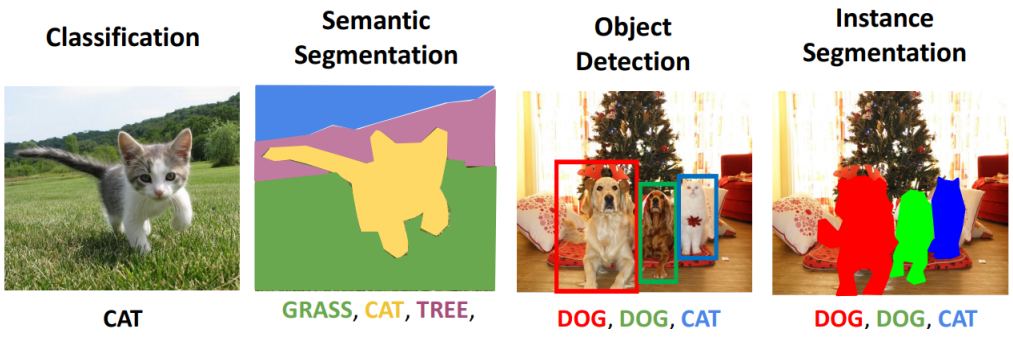

Figure 3: Different image tasks

One of the most basic and common task for computer vision are often image classification. The whole image is read by the CNNs and each image correlate to a fixed size result. The output of the CNNs will often be a vector of size N with N being the number of classes and the vector value the probability distribution of the classification result. Object detection is an extension of the image classification problem where the network is required to detect an unspecified number of objects with both the probability distribution for the object and the location of the object within the image. The two well known approaches to object detection are single state object detection and two stages object detection.

2.3.1 Single state object detection:

The commonly used single state object detection algorithm is YOLO, which stand for You Only Look Once. The image is segmented in to a S by S grid with S being one of the hyper-parameter of the algorithm. The algorithm goal is to associate each object with the box, in which its center is located. With each box in the grid, the algorithm create a fixed set of N (another hyper-parameter) anchor boxes; the anchor boxes assign to a box in the image grid will have their centers align on the box center. The output of the algorithms is a 3D matrix of shape $(S * S * (5 * N + C))$ (C is the number of classes of object and an additional class for background object).

Each vector of length $(5 * N + C)$ is the evaluation of a grid box. With each bounding box assigned to the grid box, the model will output the value of the transformation parameter so that the anchor box could match the bounding box of the object as well as a confidence score for the anchor box. The remaining C values at the end of the vector is the probability distribution for the object detected and when the result is extracted, the class with the highest value will be chosen just like in an image classification problem.

2.3.2 Two states object detection:

The common approach of two states object detection is through Regional Convolutional Neural Networks(R-CNNs) which include R-CNNs, Fast R-CNNs and Faster R-CNN. These algorithm use a heuristic methods to estimate the different regions may contain an object and then calculate the transformation parameters and object confidence score for each region. Afterward, the regions confidence score get thresholded and the regions that got accepted get put through a classification model to return the probability distribution, hence the name two states object detection.
R-CNNs use traditional computer vision techniques to propose N regions of interest (ROI) from the image data. These regions will be warped to a fixed size before passed through a CNN and then to the box transformation/confidence score prediction model. The drawback of the R-CNN approach is it time consuming training and running time due to the number of forward passes required to perform for all regions proposed by the region proposal algorithm which renders it inapplicable in a lots of industrial context. Fast

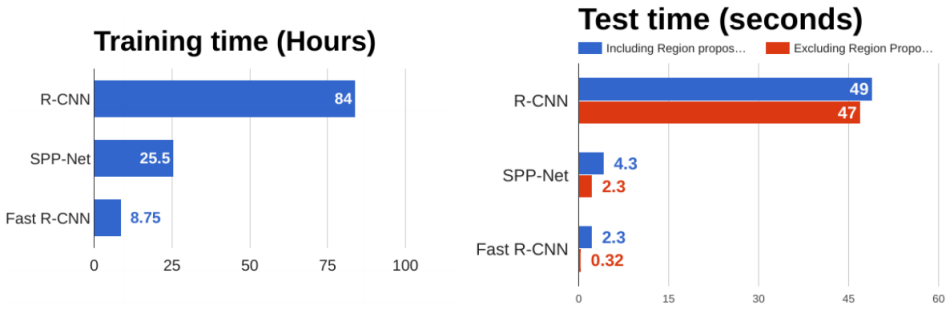

Figure 4: Train and test time of R-CNN, SPP-Net and Fast R-CNN

R-CNN solution to the problem is to run the image through a backbone

CNN network before the region proposal algorithm. The architecture of this backbone CNN can be chosen from the architectures of Resnet, Alexnet, VGG, etc. The region proposal then is run on the feature map produced by the backbone CNN, the feature regions is the cropped from the feature map and resize before passing through the network to compute the transformation, confidence score and if suitable, probability distribution over available classes. This significantly reduce the training or running time compared to R-CNN, to which the majority of time consumed is spent on the region proposal algorithm. Faster R-CNN's solution to this problem is to use a CNN

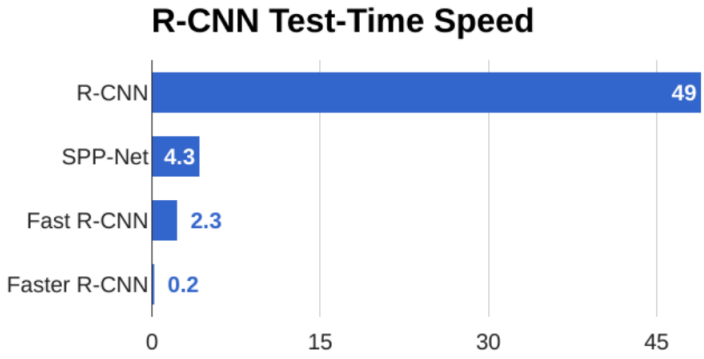

Figure 5: Test time of R-CNN, SPP-Net, Fast R-CNN and Faster R-CNN(seconds)

called the Region Proposal Network similar to YOLO algorithm to predict the regions instead of using region proposal algorithm which can be time consuming.

3 Experiment design

3.1 Exploratory data analysis

The main training data set are 120 videos of American foot ball with label bounding boxes for helmet and impact on the helmet if it occur. Extracting the video result in 26091 image frames. most of the average frame per video is 300 and the maximum number of frame is 600. The data set is imbalance with 10% number of the frame contains impact. The imbalance is even more significant at the number of helmet bounding box with impact and helmet

bounding box with none. There are only 23 helmet impact for every 1000 annotated helmet bounding box The helmet size has average area of 600 pixel area with standard deviation of 560.

3.2 Prepare data set

The given data set contains 120 videos which is splitted into train and test sets using their game key and play id on 0.7 and 0.3 ratio. Due to resources and time constraint, the data used for a given video only contains frame of the impact and the 2 neighbor frames next to it. I.e for a given impact frame there will be minimum 5 frames used in the data set and if there are 2 consecutive impact frame there will be 6 frames used in the data set. Using further frame from the impact frame will allow the model to have better result but this may not reflect the generalization of models.

3.3 Model 1: Helmet detection

Because the objective of the project value accuracy above inference time. Two stage object detector is chosen to be the object localization technique.

3.4 Model 2: Helmet impact classification

Figure 6: An example of cropped image

Both classical machine learning and deep learning are used to classify the impact of helmet. We approach the impact classification task by building model to see the feature of the environment surround the helmet during the impact. The experiment used one crop per helmet in the ground truth data set and try to classify whether this frame contains impact. Given the analysis of size in the helmet found in EDA, we use a 100x100 crop for each helmet given the helmet bounding box in the center of the image. If an image does

not fully accommodate the 100x100 crop, the crop will be resized to 100 x 100 pixels.

3.4.1 Simpler classifier for binary impact detection

3.4.2 CNN for binary impact detection

Different architecture are used for the impact classification task in which perform the best on impact data has the F1 score of 0.6.

3.5 Combined model: Impact detection using two stage detector and stand alone CNN classifier

To fully evaluated the performance of the built models on the impact detection task, result of helmet bounding box is also feed into the best CNN model mentioned above.

3.6 Another approach: Impact detection using two stage detector alone

Attemps were made to modify the architecture of the selected two stage detector models by having another "dense prediction" to increase the size of the crop feature map during the roi pool process. However due to the time constraint, this model has not been train and evaluated.

4 Evaluation

4.1 Helmet detection - Localization

A pre-trained set of weight from ImageNetPretrained data with architecture of Faster RCNN + FPN is applied. The model is converged after running 10000 epochs and produced the following result at the last epoch.

At the IoU of 0.5, the precision value is at 0.36. Generally, it is an accepted result while the cut-off value for IoU in the competition is at 0.35.

As the result, the predicted bounding boxes of the model will be served as the feed-in value in the next model.

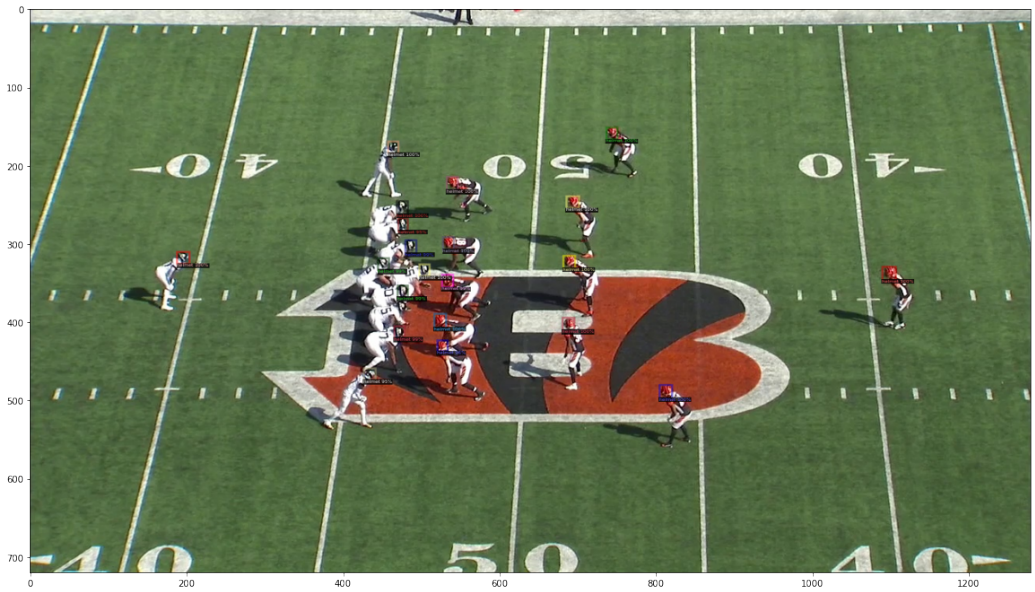

Figure 7: A sample of prediction on public test set

```
Accumulating evaluation results...
DONE (t=0.53s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all |
   maxDets=100 ] = 0.189
 Average Precision  (AP) @[ IoU=0.50      | area=   all |
   maxDets=100 ] = 0.357
 Average Precision  (AP) @[ IoU=0.75      | area=   all |
   maxDets=100 ] = 0.186
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small |
   maxDets=100 ] = 0.189
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium |
   maxDets=100 ] = 0.177
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large |
   maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all |
   maxDets=  1 ] = 0.142
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all |
   maxDets= 10 ] = 0.484
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all |
   maxDets=100 ] = 0.495
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small |
   maxDets=100 ] = 0.493
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium |
```

```
   maxDets=100 ] = 0.600
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large |
   maxDets=100 ] = -1.000
```

4.2 Classifier model - Perform Basic Machine Learning

Support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. As has been proven in many practical image classification projects, SVM can generate an acceptable result. However, the result is pretty bad under the scope of this problem. It is even not managed to predict any value of impact data points.

```
              precision    recall  f1-score   support

         0.0       0.65      1.00      0.78       592
         1.0       0.00      0.00      0.00       325

    accuracy                           0.65       917
   macro avg       0.32      0.50      0.39       917
weighted avg       0.42      0.65      0.51       917
```

Listing 1: SVM Model

Logistic regression and SGD Classifier are also applied for this problem. They performed quite well when we analyse the F1-score.

```
              precision    recall  f1-score   support

         0.0       0.69      0.76      0.73       592
         1.0       0.47      0.39      0.43       325

    accuracy                           0.63       917
   macro avg       0.58      0.58      0.58       917
weighted avg       0.62      0.63      0.62       917
```

Listing 2: Logistics Regression

```
              precision    recall  f1-score   support

         0.0       0.73      0.59      0.65       592
         1.0       0.45      0.61      0.52       325

    accuracy                           0.60       917
```

```
   macro avg       0.59      0.60      0.59       917
weighted avg       0.63      0.60      0.61       917
```

Listing 3: SGD Classifier

SGD Classifier is a bit out perform the Logistics Regression model. It will be presented to perform on hidden test set.

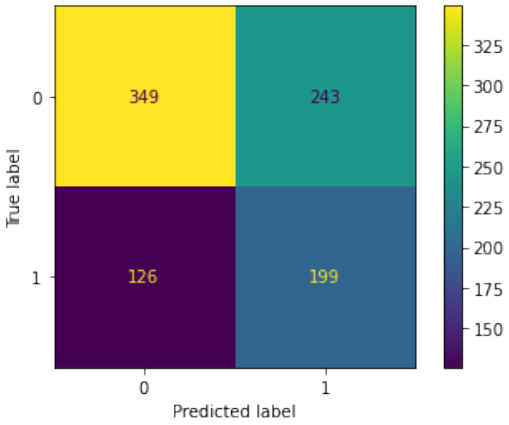

Figure 8: Confusion matrix of SGD Classifier

4.3 Classifier model - Apply Artificial Neural Network

Image classification is one of classical problems of concern in image processing. Using neural network is proven to be useful in a lot of research papers.

At this step, a simple architecture of ANN structured. The summary of model is as in the Listing 4:

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0,
   dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=400, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=84, bias=True)
  (fc3): Linear(in_features=84, out_features=32, bias=True)
  (fc4): Linear(in_features=32, out_features=2, bias=True)
)
```

Listing 4: Neural net architecture

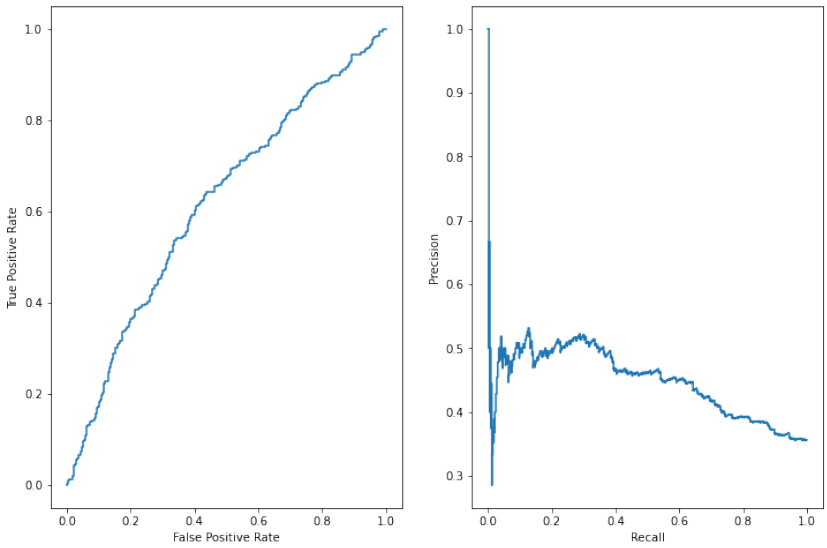

Figure 9: ROC curve and Pression-Recall curve of SGD Classifier

```
                precision    recall  f1-score   support

             0       0.79      0.68      0.73       592
             1       0.54      0.67      0.60       325

    accuracy                           0.68       917
   macro avg       0.66      0.68      0.66       917
weighted avg       0.70      0.68      0.68       917
```

Listing 5: Neural Network Classification Report

4.4 Classifier model - CNN architecture

Input images CNN are normalize and resize to 100x100 pixel before feed into the CNN. In addition, data augmentation techniques such as rotate, flip and helmet occulusion are also used to help model generalize. The best classification score for used CNN architectures in the project was squeeze and excitation CNN with aggregated residual (Se_ResNext50_32x4d). This architecture achieves better performance than others residual network with the same size and smaller simple CNN build from scratch.

```
                precision    recall  f1-score   support

```

```
              0       0.88      0.92      0.90     10128
              1       0.62      0.51      0.56      2532

    accuracy                           0.84     12660
   macro avg       0.75      0.72      0.73     12660
weighted avg       0.83      0.84      0.83     12660
```

Listing 6: CNN Classification Report

4.5 Further implementation

Under the a tight time constraint, many approaches and extended methods have not been applied yet. In the next few days, the notebooks will be adjusted for competition submission. All the evaluations above are generated from subset of ground truth and the unseen data set has not been tested.

In the next courses in Deep Learning and Computer Vision, the problem will be further studied in following scopes:

- Applying two stages detection.
- Applying object tracking to synchronize two views.
- Utilize the sensor data.
- Video classification using 3D CNN, techniques to take advantage of the time domain.