Real-Time-PPE-Detection with-YOLO

Mustafa Tayyip Bayram 12237686

Table of Contents

[ABSTRACT 2](#_Toc128263357)

[1. Motivation 2](#_Toc128263358)

[Methods and Obtained Results 3](#_Toc128263359)

[2. Data 3](#_Toc128263360)

[3. Theoretical part 6](#_Toc128263361)

[Related Studies 6](#_Toc128263362)

[Why and What is YOLO? **Error! Bookmark not defined.**](#_Toc128263363)

[Performance Measure **Error! Bookmark not defined.**](#_Toc128263364)

[General Pipeline **Error! Bookmark not defined.**](#_Toc128263365)

[Hyperparameters **Error! Bookmark not defined.**](#_Toc128263366)

[Expected Results **Error! Bookmark not defined.**](#_Toc128263367)

[4. Implementation **Error! Bookmark not defined.**](#_Toc128263368)

[5. Evaluation and Conclusion 10](#_Toc128263369)

[6. Future Steps 11](#_Toc128263370)

[References 11](#_Toc128263371)

## List of Figures

[Figure 1. Top causes of construction related causalties 2](#_Toc128240830)

[Figure 2. Image example Figure 3. Label file example 4](#_Toc128240831)

[Figure 4. Label Instances wandb 5](#_Toc128240832)

[Figure 5.mAP **Error! Bookmark not defined.**](#_Toc128240833)

[Figure 6. Pipeline](#_Toc128240834)

[[Code 1. Retraining snippet](#_Toc128240834) **[Error! Bookmark not defined.](#_Toc128240834)**](#_Toc128263273)

[[Code 2. Detection on youtube video](#_Toc128240834) **[Error! Bookmark not defined.](#_Toc128240834)**](#_Toc128263274)

**[Error! Bookmark not defined.](#_Toc128240834)**

[Figure 7. yolov5 vs yolov7 **Error! Bookmark not defined.**](#_Toc128240835)

[Figure 8. yolov5s vs yolov5x **Error! Bookmark not defined.**](#_Toc128240836)

## ABSTRACTYarak

PPE (Personal Prospective Equipment) detection in real time with object detection YOLO is a system that uses deep learning algorithms to recognize items automatically in real time. This technique is especially beneficial in situations where object detection is vital to guaranteeing individual safety, such as building sites or manufacturing plants.

Real-time PPE detection with YOLO takes advantage of this algorithm to detect PPE in real-time. This includes detecting items like hard hats. In this report, just hard hats are trained and others equipment’s such as vest, gloves, googles can be extended with required data. The system might use a camera to capture the stream of the workers, and the YOLO algorithm is applied to the stream to detect any PPE that is present or not.

Links

GitHub Repository: <https://github.com/mutabay/Real-Time-PPE-Detection-with-YOLO>

Kaggle Data: <https://www.kaggle.com/datasets/mustafatayyipbayram/ppe-detection>

Main Notebook: <https://colab.research.google.com/drive/1J-bNoEjOz-Ivqj5GPpYumExYfo48-CEK?usp=sharing>

Wandb results for YOLOv5: <https://wandb.ai/mutabay/YOLOv5>

Wandb results for YOLOv7: <https://wandb.ai/mutabay/YOLOR>

All Files Link: <https://www.filemail.com/d/gfmjthryxfqlffj>

## Motivation

The motivation behind personal protective equipment (PPE) detection is to improve workplace safety by ensuring that employees are wearing the necessary equipment to protect them from hazards. The top cause of construction related casualties are falls, people getting stuck in the equipment, collisions. It can be prevented if workers wear appropriate personal prospective equipment’s like vest, googles, hard hats etc. The issue with PPE is that it is frequently required for workers to wear it, but it can be uncomfortable, difficult to wear for long periods of time, and easily forgotten or misplaced. The failure to wear the proper PPE can result in serious injuries or even death in some cases.



Figure 1. Top causes of construction related causalties

Detecting PPE in real time using object detection is a significant solution to this issue. Employers may track and monitor the use of PPE in real time, ensuring that employees are wearing the proper equipment and boosting workplace safety. The technology is especially important in high-risk situations like construction sites, manufacturing plants, and healthcare facilities. In this project, since there is a problem of finding data especially labelled, construction site hard hat detection has been applied.

### Methods and Obtained Results

In general, there are several approaches to resolving this problem. As an example, adding numerous sensors to each PPE and providing control for all of it. If we take a single business area as an example, then thousands of sensors would be required otherwise. As a result, using computer Vision technology will be significantly less expensive, more environmentally friendly, and produce more accurate findings.

This problem was precisely addressed in this paper by adopting the YOLO architecture. These models can then be integrated with CCTV cameras and specific triggers in the following stage of the project to provide results.

Different YOLO models and versions were utilized in order to improve the result and obtain different findings. Standard models yielded more optimized results, as large and complex models require a lot of resources for real time checking. As a result, YOLOv5s gave the best result regarding to quickness to capture objects and accuracy. 0.92 mAP (mean Average Precision) result received with that result. According to the available sources, such as training environment, data, and workload, it is possible to state that it is a decent result.

It is feasible to claim that the most significant and risky aspect of the project is gathering adequate data and cleaning it. The data was collected from open source and cleansed adequately. Detailed information about the data is given below.

Finally, interpretations and reports were created using the necessary tools. These tools are Wandb, TensorFlow analysis tool.

## 2. Data

Roboflow Hard Hat Workers Dataset <https://public.roboflow.com/object-detection/hard-hat-workers>

Data consists of jpg format images and labels that belongs them. Every image has a special txt file that specifies which object it is were with coordinators. The formatting is organized based on the YOLO requirement. There are no missing values, and all images have representative files.

 

Figure 2. Image example Figure 3. Label file example

The data has some drawbacks. Most important one is that images are not realistic case of use case of this project for real life. The majority of the images are close-ups. If we had data labeled from afar and from those angles, as in CCTV cameras, we could create much more realistic situations and train models on this accurate data. Eventually, CCTV camera detections would be significantly more accurate. Since none of the agents (cameras) will ordinarily detect a worker in front of their face.

|  |  |  |  |
| --- | --- | --- | --- |
| **Shared By** | **License** | **Publication Date** | **Data Count** |
| Northeastern University – China | Public Domain | September 2022 | Totally 7035 Splitted Images |

|  |  |  |
| --- | --- | --- |
| **Train** | **Test** | **Validation** |
| 3688 | 1581 | 1766 |

|  |  |  |
| --- | --- | --- |
| Labels | | |
| Head (Non-Hard Hat) | Hard Hat | Person (Removed) |

*Chart, bar chart

Description automatically generated*

Figure 4. Label Instances wandb

The amount of data is not little, but if it were a little larger, it would contribute more to training. In terms of label balance, the Person label was present in very few images (about 30), therefore cleaning it made more sense because it disrupted the balance significantly. In overall, the data does not appear to be especially balanced. The figure above shows that there are 4000 Head instances and 14000 hard hat instances. To improve quality, it would be appropriate to balance the number of labels.

Data Augmentation is possible with an open-source project for YOLO. It didn’t apply in this project but can be used for next versions.

Open-Source Project: <https://github.com/srp-31/Data-Augmentation-for-Object-Detection-YOLO->

## Theoretical part and Implementation

The multi-label movie genre classification using movie plot data learning task is to predict the genres of a movie based on its plot synopsis. This is a supervised learning challenge in which a machine learning model is trained on a dataset of movie narrative summaries and the genre classifications that are associated with them. The model must learn to identify the most relevant genres for a given plot summary and classify the movie accordingly.

In this task, each movie may belong to multiple genres, so the model must be able to predict multiple labels for each instance. For example, a movie may be classified as both a romantic comedy and a drama. This makes the task more challenging than traditional single-label classification problems, where each instance is assigned a single label.

To perform multi-label movie genre classification, various machine learning *and deep learning* algorithms can be used, such as logistic regression, decision trees, or neural networks. The performance of the model is typically evaluated usin*g micro f1 score, accuracy, and recall.*

*There are numerous studies on this subject in general. Several algorithms and data sets are discussed, and the results are compared. This research is crucial in the development of the algorithms discussed below.*

### Tools

|  |
| --- |
| Data Collection Platform: KAGGLE |
| Algorithms: LOGISTIC REGRESSION (TFIDF, TFIDF + TOPIC MODELLING), NAIVE BAYES (TFIDF), NN |
| Implementations Platform: Dataspell |
| Evaluation Tools: Wandb |
| BIG LIBRARIES: KERAS, MATPLOTLIB, SKLEARN, TQDM, NLTK, TENSORFLOW |

### Related Studies and Applications

<http://cs230.stanford.edu/projects_fall_2021/reports/102983714.pdf>

<https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/>

<https://medium.com/analytics-vidhya/machine-learning-multi-label-classification-mpst-movie-plot-synopses-with-tags-tags-8314e6841e17>

<https://www.arxiv-vanity.com/papers/1801.04813/>

### Data Preprocessing

In order to train a machine learning or deep learning model, the raw data must be transformed into a format that can be used for multi-label movie genre classification utilizing movie plot data. Because the performance of the final model is hugely affected by the quality of the input data, the data processing stage is crucial.

Typically, movie plot summaries and their related genre labels make up the raw data for this activity. Preprocessing and cleaning text data to get clear of irrelevant stuff such stop words, punctuation, and special characters is typically the first step in data processing.

Graphical user interface, application, PowerPoint

Description automatically generated

Figure 5. Data Preprocessing Pipeline

In summary, data processing in multi-label movie genre classification involves cleaning and preprocessing text data, transforming it into a numerical representation, and splitting the data into training, validation, and test sets.

In conclusion, text data must be cleaned up, preprocessed, converted to a numerical representation, and divided into training, validation, and test sets before being used in multi-label movie genre classification.

### Text Featurization

Text featurization is the process of transforming textual data such as movie plot summaries into a set of numerical features that can be fed into a machine learning model as input. This is done in the context of multi-label movie genre classification utilizing movie plot data. The goal of text featurization is to extract meaningful information from text data and represent it in a way that can be easily understood and processed by machine learning and deep learning models.

There are several different text featurization techniques, each with advantages and disadvantages. Such as: Bag-of-words, TF-IDF, Word Embeddings.

#### Word Embeddings

Each word in the corpus is represented as a vector in a high-dimensional space, with semantically comparable terms clustered together. The weighted average of a document's components word embeddings is then used to represent it.

#### TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF (Term Frequency-Inverse Document Frequency) is a natural language processing (NLP) text featurization technique that gives weights to each word or term in a document based on its frequency in the document and rarity in the total corpus of documents.

In multi-label classifications, the text featurization TF-IDF (Term Frequency-Inverse Document Frequency) is frequently used. TF-IDF was a great choice for us because it captures word importance basically and other reason was TF-IDF is a relatively simple and efficient method for text featurization. It requires minimal preprocessing and is easy to implement.

##### What is N-Gram?

An n-gram is a contiguous sequence of n items (usually words) from a given document in TF-IDF text featurization. Depending on the desired granularity of the analysis, the value of n in an n-gram can range from 1 to any integer number.

An n-gram is an expansion of the concept of a "word" as a linguistic unit that can capture local contextual relationships between words in a document. For example, the word "machine learning" can be represented as a 2-gram, with "machine" and "learning" treated as separate elements in the sequence. In the project, it employed from 1 to 4 grams.

#### Topic Modelling – LDA (Latent Dirichlet Allocation)

Another text featurization technique often employed in multi-label classification is topic modeling, specifically Latent Dirichlet Allocation (LDA).

LDA is a probabilistic generative model that represents documents as a collection of topics, with each topic representing a distribution over words. The purpose of LDA is to learn the latent topics that underpin the observed documents, as well as how each subject is distributed across the corpus. The LDA algorithm accepts a corpus of documents and returns a set of topic distributions, with each document represented as a probability distribution over the subjects.

LDA can be used to extract the latent topics present in movie plot summaries in the context of multi-label movie genre classification. Each movie plot synopsis is represented as a collection of subjects, each with its own probability distribution over words. The generated topic distribution can be used to train a multi-label classification model by using it as a set of features for each movie.

### Modelling

There are numerous machine learning and deep learning models that can be employed for the task. However, we employed different models for different reasons.

#### Logistic Regression

Logistic Regression is a popular approach for binary classification tasks, in which the goal is to predict whether or not a sample belongs to a specific class. However, logistic regression can also be utilized for multi-label classification problems, such as classifying movie genres using movie plot data.

One of the benefits of logistic regression is its ease of use and interpretability. Based on the input features, it is a linear model that calculates the likelihood of a sample belonging to each label class. The model makes a binary choice for each label class based on a threshold value that can be changed to regulate the precision/recall trade-off. Different thresholds are used => [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9] Best results are taken by 0.4 and 0.5 threshold.

Logistic Regression is used on the project because it works well with high-dimensional data. Detailly logistic regression is computationally efficient and can handle large datasets with a large number of features. It can be easily extended to handle multi-class and multi-label classification tasks, and can be used with One-vs-Rest technique.

In the project, logistic regression used with TFIDF technique and TFIDF + Topic Modelling (LDA) methods, then results are compared.

#### Naive Bayes

The Naive Bayes algorithm is a probabilistic technique that is commonly used for text classification tasks, like our task. Naive Bayes implies that the features are conditionally independent given the class label and uses Bayes' theorem to estimate the probability of each label class given the input features.

For text classification, Naive Bayes serves as a benchmark and because of that it is recommended that this model be used. Naïve Bayes is used with TFIDF.

#### Neural Network

At first, validation data and training data merged for training. Deep learning is an extremely strong tool for data featurization. So, we experimented with it as well. Long Short Term Memory networks, sometimes known as "LSTMs," are a type of RNN that can learn long-term dependencies. The sequence information is handled by LSTM.

At the beginning, Tokenizer used from the keras library to preprocess the text data. Then, it fits the tokenizer on the training data to create a vocabulary and assign a unique integer to each word in the corpus. Then, it converts the text data to sequences of integers. Later it's ensured that all the sequences have the same length by padding or truncating them.

About the model:



Figure 6. NN Model

##### Hyper Parameters and Selections

1. An Embedding layer, which maps the integer-encoded words to dense vectors of fixed size. This layer has vocab\_size as its input dimension, output\_dim of 50 and an input\_length of 1200.
2. An LSTM layer with 128 units and returns the output for each time step in the input sequence (i.e., return\_sequences=True).
3. A Dropout layer with a rate of 0.5 to prevent overfitting.
4. Another LSTM layer with 64 units.
5. Another Dropout layer with a rate of 0.5.
6. A Dense output layer with 71 units and a sigmoid activation function, which produces a binary classification output.
7. Batch Size = 16, as Model created in local, we couldn’t give much.
8. Epoch number = 16
9. As an optimizer, Adam is used.



Figure 7. Model Fitting

## ****Evaluation**** and Conclusion

Results are available on wandb.

<https://wandb.ai/mutabay/movie_genre_classificaiton?workspace=user-mutabay>

Graphical user interface

Description automatically generated Chart, line chart

Description automatically generated

Figure 8. NN Results

According to the visuals, max F1-measure is 0.3069 when threshold is 0.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic Regression + TFIDF | Logistic Regression + TFIDF + Topic Modelling (LDA) | Naïve Bayes + TFIDF | Neural Network |
| Micro F1 Score | 0.49 Train  0.39 Test | 0.55 Train  0.34 Test | 0.33 Train  0.25 Test | 0.33 Train  0.31 Test |

Other metrics are available on wandb. However, as Micro F1 is our performance measurement, it is more convenient about giving insights.

Eventually.

* Logistic Regression with TFIDF gave the best result and it is the most effective as it is simplest architecture.
* Logistic Regression with TFIDF and Topic Modelling (LDA) gave a bit overfitted result and it should not preferable than before.
* Naïve Bayes + TFIDF is also very simple model but much. It gave the worst result as we already created it to take it as base value.
* We were hoping to get better results with Neural Network but as we tried some other hyperparameters, this one was one of the best. After making more epoch number, it might be better.

## ****Future Steps****

* Data augmentation can be used to diversify data and better recognition.
* More suitable data can be use, more balanced, more realistic regarding to use it in CCTV cameras. (Perspectives and angles)
* Data number should be more for getting better accuracy results.
* Data labels can be increased, for example adding google detection extension will be more useful for general solution.
* Training can be made with better resources, especially required for X models.
* Try new hyperparameters.

# **References**

|  |  |
| --- | --- |
| [1] | Z. Hassan, "Instance Segmentation Vs. Object Detection: 3 Things You Need To Know folio3.ai," 2022. [Online]. Available: https://www.folio3.ai/blog/instance-segmentation-vs-object-detection/#:~:text=So%2C%20the%20difference%20between%20instance,instance%20present%20in%20visual%20input.. |
| [2] | R. Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms," 2018. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. |
| [3] | S.-H. Tsang, "Review: SSD — Single Shot Detector (Object Detection)," 2018. [Online]. Available: https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11. |
| [4] | M. Rizwan, "YOLOv7 and YOLOv5 Comparison on Embedded Devices and Computer Systems | cameraanalyze.co," [Online]. |
| [5] | M. R. Munawar, "How do Hyperparameters of YOLOv5 Work? medium.com," 2022. [Online]. Available: https://medium.com/augmented-startups/how-hyperparameters-of-yolov5-works-ec4d25f311a2. |
| [6] | ultralytics, "About hyperparameters," 2022. [Online]. Available: https://github.com/ultralytics/yolov5/discussions/7319. |