AN MULTI-DIMENSIONAL VIDEO REVERSE SEARCH ENGINE USING COMPUTER VISION AND MACHINE LEARNING

Qiantai Chen[1] and Yu Sun[2]

[1]Department of Computer Science
University of California - Irvine, CA 92697
[2]Computer Science Department
California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Online media has become a mainstream of current society. With the rapid development of video data, how to acquire desired information from certain provided media is an urgent problem nowadays. The focus of this paper is to analyse a sufficient algorithm to address the issue of dynamic complex movie classification. This paper briefly demonstrates three major methods to acquire data and information from movies, including image classification, object detection, and audio classification. Its purpose is to allow the computer to analyse the content inside of each movie and understand video content. Movie classification has high research and application value. By implementing described methods, finding the most efficient methods to classify movies is the purpose of this paper. It is foreseeable that certain methods may have advantages over others when the clips are more special than others in some way, such as the audio has several significant peaks and the video has more content than others. This research aims to find a middle ground between accuracy and efficiency to optimize the outcome.

KEYWORDS

Convolutional Neural Network, Image Classification, Object Detection, Audio Classification, Movie Classifier.

1. INTRODUCTION

The topic of video classification has achieved outstanding results in recent years, and diverse ways of approach such as implementing complex algorithms and architecture improved the overall accuracy in a certain genre of video [11][12]. By making the computational device understand and recognize the video content is a major challenge for nowadays technology applications. As the increasing amount of contribution has been done in this area, it is more often to recognize the hidden value for the video classification in both commercial and research areas. The usage of identifying the video content can be employed in numerous applications including autonomous driving, rocket science, internet streaming, home assistant, and analyzing sports video.

One of the areas that desperately needs the video classification is the online streaming application such as YouTube [13] and Netflix [14]. There are approximately more than five hundred of videos uploaded to YouTube every minute. The methods of automatically diagnosing and analyzing the language information, understanding the content, captioning the video with

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 125-136, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120911

categories and description, and coming up with relatively high accuracy compared with humans is a classic concern. On the other hand, image classification not only contributes to the entertainment application but also makes autonomous driving become a reality. There are several existing vehicle models capable of self-driving have been announced around the world for a while now, and some of the brands are really achieving some significant results in this area by utilizing the video classification real time with live feedback.

The traditional approach of video classification is that of creating a decision tree [1][2][3]; many researchers and research papers indicate that there is still a great amount of potential improvement leftover for decision tree's accuracy. Some methods include expanding the size of the dataset as well as combining several layers of discrete spectral information. However, with the rapid development of deep learning style machine learning [15], especially by utilizing the approaches including convolutional neural network, long short-term memory, and gated recurrent unit, modern artificial intelligence outpaces the traditional approaches in both commercial and academic use. These approaches successfully demonstrate significantly greater efficiency and more wide-spread application than decision trees. Some of the research publications on the topic offer even more specific approaches that utilize these modern deep learning techniques. It is common to acknowledge that object grounding is relatively mature technology and also has greater potential when used for both commercial and government areas. There are multiple ways to play around with object grounding [4] and the way is by using the different settings within the models including Zero-Shot Grounding-Net [5] and Video Object Grounding-Net [6]. On the other hand, in one of the research findings, which shows there is a significant margin when they are trying to identify the tennis videos with the sound [7]. In between classifying the video, looking for a potential existing paddling sounds that only could exist through playing tennis could quickly help the researcher to locate the desired frame. Another significant way of identifying the video content is by utilizing the Video Temporal Analysis, there are a lot of exciting methods underlying such as temporal activity detection, language-based video search, and action anticipation, but it is still an interesting topic to unraveling the mysteries due to the reasons that great number of videos has such complex temporal structures, great video variation and problem scale.

The methods that have been conducted in this research were those of creating a total amount of three coding programs to achieve a different result, since the problem set requires a variety of algorithms to solve it. An analysis will be conducted after implementing all three programs. The other method is by gathering the existing programs, and creating a complex algorithm that utilizes pre-existing results for further computation. The research goal is to examine the best feature extraction method and optimize both the effectiveness and efficiency of the program. The part of the program where we implemented the audio classification approach is inspired by the research paper written by Xun Gong and Fucheng Wang [7], who indicate there is a significant improvement in their program by identifying the sounds of hitting a tennis ball. The idea of selecting certain peaks within the voice and match noise, as well as the methods in which they did this, enlightened our thoughts of something similarly involving key film noise that appear in certain movie genres, such as explosions in the action genre. By considering this idea, searching for desired sounds gives the program ability to quickly eliminate a great portion of the sample data. There are some good features we take to achieve our result! Our approach utilizes multi-layered programming techniques, a giant database for object detection and recognition, interactable human interface design, and the ability to present an option for users to upload movies that are currently not listed in our database, thus updating the model further.

There are several different measurements implemented to ensure the results are following the prediction that we are expecting. The way to prove our result is by utilizing cross-validation in the design process, so that the program will automatically emphasize a confidence level of the accuracy that these certain methods could eventually end up with. Meanwhile, the program of image classification uses application of neural networks, which utilizes the validation and testing datasets to determine if the processed results match with our two other additional datasets. To further mitigate the possibility of edge case, the program also has the ability to compare the model accuracy and select the highest accuracy among all of the models. There is also an option by doing it manually is to customize the model using hyper parameter tuning. Once our analysis of the results outlines recommended models, we select one for each area and move on to the next step of tuning the models in the system. From there it is easy to determine what changes are needed for our general heuristic and to get the highest accuracy possible for the combined system. The final stage for our program is by conducting a comprehensive case study on the system, and analyzing the statistical data to determine if it still meets both the standard of effectiveness and efficiency.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, followed by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In this section, we layout the key technical and research challenges to address.

Challenge 1: Maintaining project speed across multiple machine algorithms is inherently difficult. Even just one classification machine learning algorithm could consume a great amount of time and processing resources. With the time and space complexity of just one, trying to manage a reasonable speed in between the three identification methods that this paper is trying to imply is difficult. Given the complex nature of machine learning algorithms, it is important to maintain them across systems and limited computational resources. Therefore, parallel processing is the most efficient choice, especially when facing the design that algorithms should run independently without interaction or exchanging of information. The general solution is that we collect the results three times from each individual algorithm and then use a supplemental program to pick the final prediction with the highest confidence from the results generated. Parallel processing not only allows us to achieve a relatively high accurate result, but also presents the ability of manipulating the algorithm’s speed in different circumstances. In terms of accuracy, parallel processing also allows for cross-referencing of the individual models’ results as opposed to just picking the highest confidence. This allows for a decently robust solution.

Challenge 2: Selecting what to sample and how much of it to sample including frequency and length can lead to vastly different results. Selecting a database from the original sources could result in an insufficient amount of data or overwhelming unnecessary computation. Small changes to training or testing data can result in widely varying accuracies due to the nature of classification and other model designs. Movies, which are usually hundreds of gigabytes per film, are especially an issue when you take into account the number of movies that would need to be preprocessed for such a project. The concept for optimizing the right amount of data is usually the hardest thing to do given consideration for both selecting data now as well as updating it in the future with the consistent release of films. Videos themselves can also vary considerably in quality even from the same footage of the film, so it can drastically affect predictive performance. Generally, the particular solution is that we process every frame of the movie so that when the algorithms come in, we will always gain the best dataset. In this approach, we sacrifice the calculating power and receive a relatively complete set of data. By ignoring certain frames of the video, such as the beginning of the movie where it usually just starts with the black images, movies usually take several frames to switch from one scene to another. One way of mitigating the general solution's risk is by measuring the average change of scene and rescale it under different genres or categories of a movie. Meanwhile, to prevent the inaccurate dataset being processed without our knowledge, a validation dataset and training dataset has been set up to make sure the result is exactly what we expected.

Challenge 3: Many machine learning models are better suited for individual tasks. For example, image classification often runs better using Neural networks than standard classification algorithms. This variance forces researchers to study different potential solutions to find the optimal one. There are so many existing models that have been created and published to the world, and only few of them are worthy to investigate deeper into it, as the nature of the concerns in mathematical, statistical, and computational fields are different in each model. How to decide the best model is definitely one of the hardest parts of the whole experimenting process, since we are just unable to try all of the published data models. Our problem requires us to not just go through the process of selecting an individual model, but doing this three times for three entirely different subjects. The general solution is by going through a selection of models or just randomly picking a common machine learning model which is highly recommended at the time. You then run a few tests to generate the result but not enough to require into the greater depth and understanding of each model and its algorithm. This paper is doing research based on other researchers' results, and coming up with the machine learning model that is already working. We are not only examining the validity of the model but also selecting the model that has proven most useful when applied to similar problems.

3. Solution

The specific solution to build the system and address the challenges above will be presented in this section.

3.1. Overview of the Solution

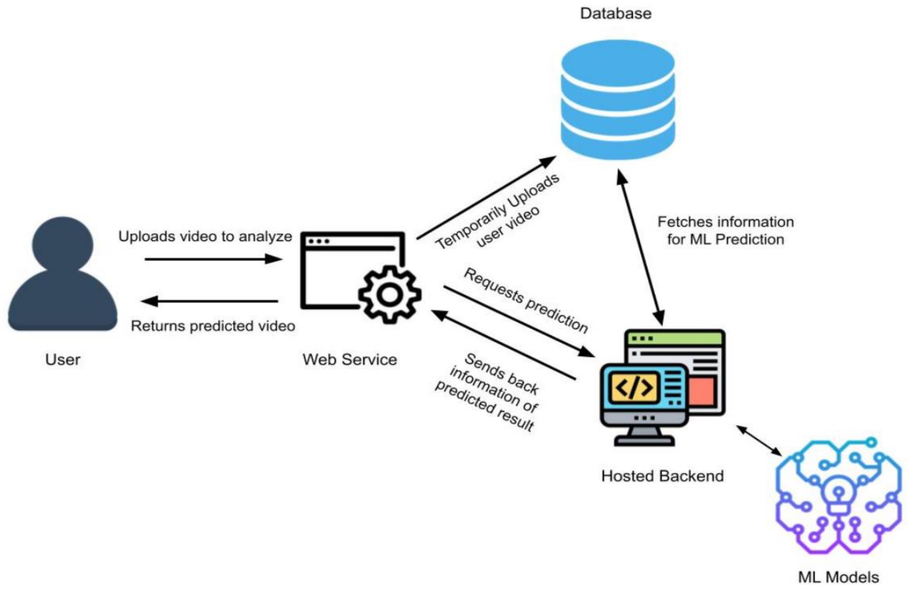


Figure 1. An Outline of the Video Prediction Web Service

The user has the option to upload a video onto a website page, where it connects to both the database and backend system. The interaction between user and system is solidly limited to the web service front end that we implemented only for proper user interaction, although the web service page will eventually return a predicted movie name back to the user. There are several processes going on behind the scenes of our web service page. The web services will transfer the user uploaded video to the database for temporary storage, and access the hosted backend to request a prediction for the temporary video clip. The clip will be accessed later on by the backend and machine learning portions of the application. The step is starting with the database, which passes the movie clip uploaded from the user as an output, and the hosted backend will accept the movie clip as input for further evaluation and process. Once the video has been processed, the machine learning models involved in the program will process the movie clip with each of the different methods and tools to analyze what film it may have originated from. Once the movie clip is processed, the result will transfer back to the backend, which will access the pre-processed original movie dataset stored in the database. After finalizing the result coming out of the system, the hosted backend will essentially present a piece of information regarding the final predicted result from the system back to the web service page, which is the final stage for the whole system, displaying the end result for the user.

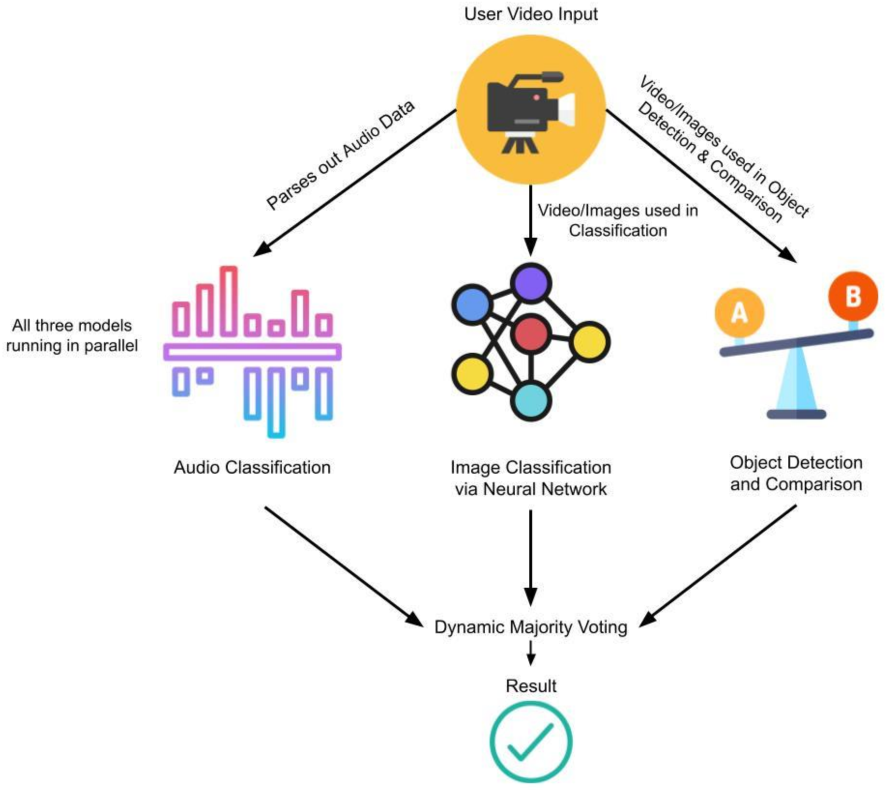
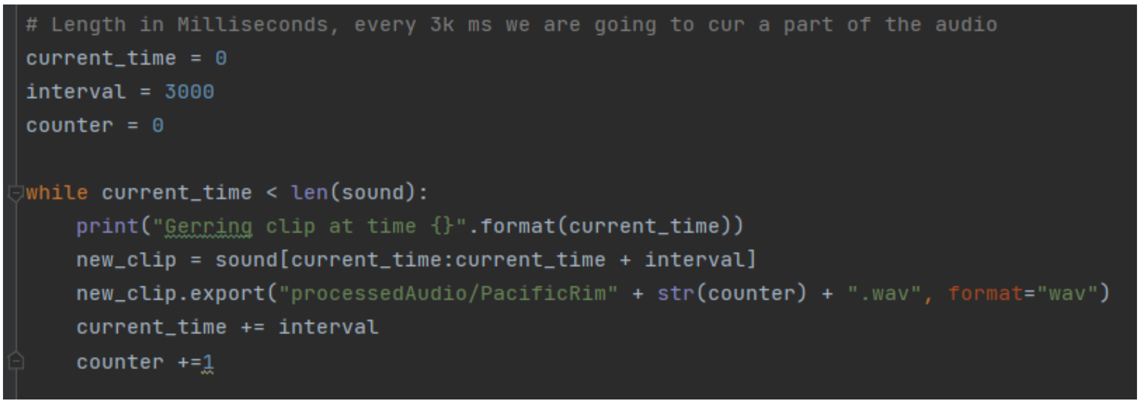

User Video Input
Parses out Audio Data
Video/Images used in Classification
Video/Images used in Object Detection & Comparison
All three models running in parallel
A
B
Audio Classification
Image Classification via Neural Network
Object Detection and Comparison
Dynamic Majority Voting
Result

Figure 2. Outline of the custom machine learning process

3.2. Audio Classification

The audio classification is one of the most mature techniques in nowadays, there are many audio streaming companies implement a similar build in functionalities in their application. The hardest part of this research is to determine how we want to extract the audio from either database or user upload. The following code segment emphasis the idea of how we decide to select the audio, and deliver the audio part for further analysis. Conversion from a .mp4 file to .mp3 file is first step needs to be processed. Every 3 thousand millisecond an audio segment will be selected and converted into a .wav file, which is much easier for computational device to do the machine learning. The algorithm will do the segment-level feature extraction, which gets feature matrix for each dataset. When the feature matrix successfully extracted, the algorithm can perform machine learning including cross validation and convolutional neural network.



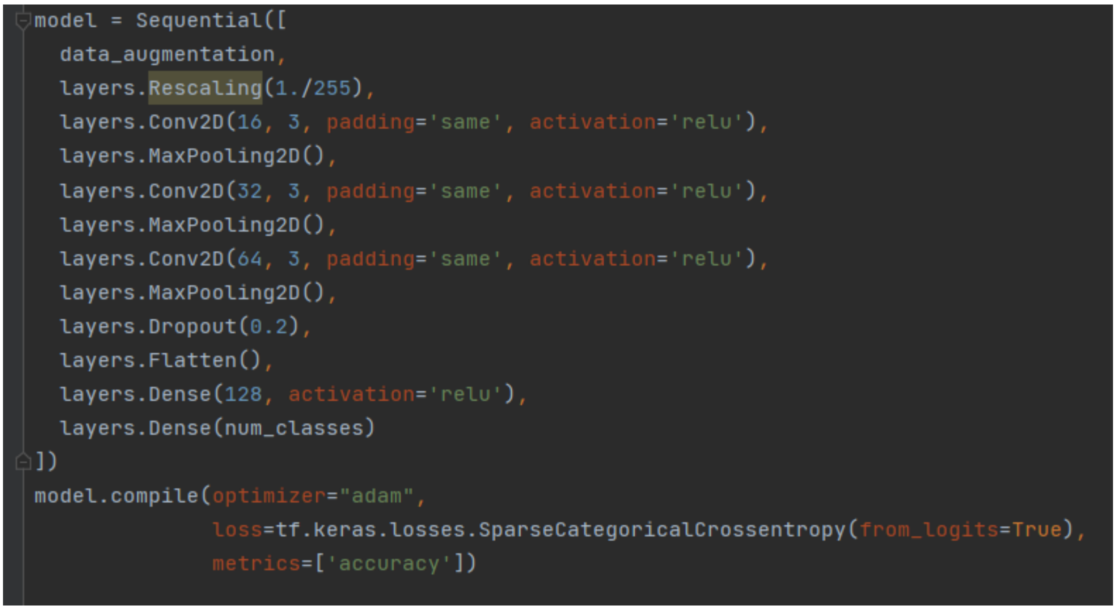
```python
# Length in Milliseconds, every 3k ms we are going to cur a part of the audio
current_time = 0
interval = 3000
counter = 0

while current_time < len(sound):
    print("Gerring clip at time {}".format(current_time))
    new_clip = sound[current_time:current_time + interval]
    new_clip.export("processedAudio/PacificRim" + str(counter) + ".wav", format="wav")
    current_time += interval
    counter +=1
```

Figure 3. The code excerpt of the custom audio file process

3.3. Image Classification

Since the application is expecting database in great scale, the first thing is to shrink however the database it has to exactly what we require to do the prediction, and avoid unnecessary computer computing recourses to decrease the efficiency of algorithm. The algorithm is already processed the image by rescaling the frame, taking screenshots of data in every three seconds, grey-scaling the image, and categorize the dataset. Meanwhile, the algorithm also initiates a validation dataset, testing dataset, and training dataset. Each dataset creates for purpose of providing dataset for algorithm self-predicting, saving original database for future reference, and getting ready for convolutional neural network machine learning respectively.



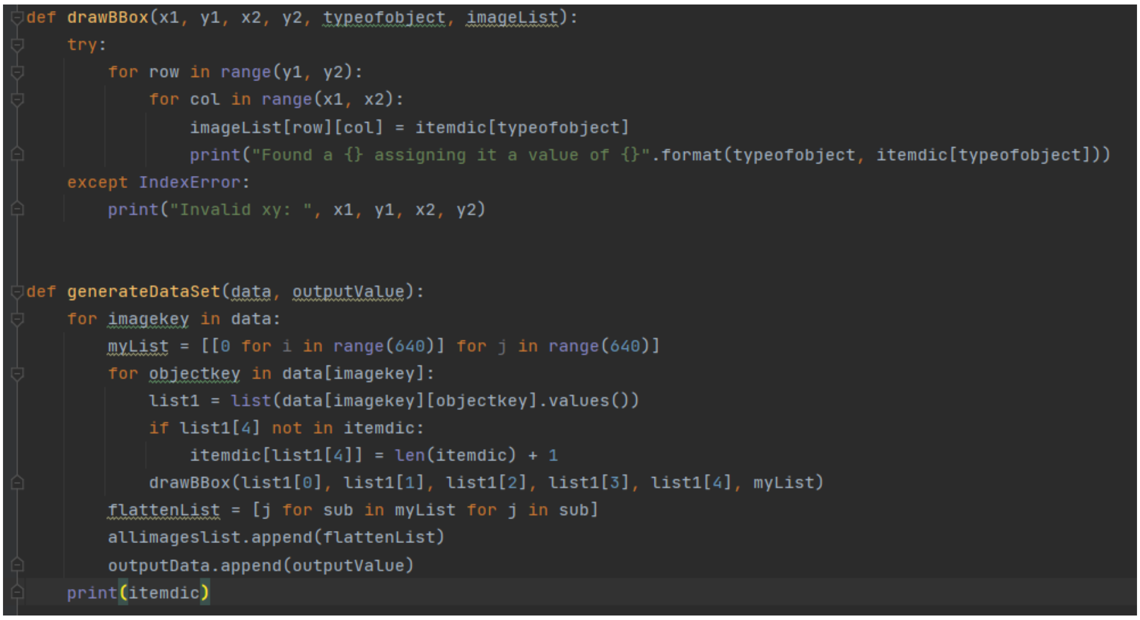
```python
model = Sequential([
  data_augmentation,
  layers.Rescaling(1./255),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
model.compile(optimizer="adam",
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure 4. Outline of the custom machine learning process

The above code segment is designed to create the model, which consists of three convolution blocks with a max pool layer in each of them. There's a fully connected layer with 128 units on top of it that is activated by a *relu* activation function. In order to address the overfitting issue, the application automatically eliminate duplicate frames.

3.4. Object-based Classification

By utilizing a public object detection architecture named Yolov5 and modify behavior of the code, we are able to detect object position with each of the frame and stored the information in a .json file. The coordinate of the boxing boundaries and type will be placed in .json file. As long as we have the analyzation for each frame, we are able to place each .json file into a new .json file which combined all frame analyzation for one movie. At this point, each movie will have an individual .json file, and its ready to do next step.



```
def drawBBox(x1, y1, x2, y2, typeofobject, imageList):
    try:
        for row in range(y1, y2):
            for col in range(x1, x2):
                imageList[row][col] = itemdic[typeofobject]
                print("Found a {} assigning it a value of {}".format(typeofobject, itemdic[typeofobject]))
    except IndexError:
        print("Invalid xy: ", x1, y1, x2, y2)


def generateDataSet(data, outputValue):
    for imagekey in data:
        myList = [[0 for i in range(640)] for j in range(640)]
        for objectkey in data[imagekey]:
            list1 = list(data[imagekey][objectkey].values())
            if list1[4] not in itemdic:
                itemdic[list1[4]] = len(itemdic) + 1
            drawBBox(list1[0], list1[1], list1[2], list1[3], list1[4], myList)
        flattenList = [j for sub in myList for j in sub]
        allimageslist.append(flattenList)
        outputData.append(outputValue)
    print(itemdic)
```

Figure 5. The code excerpt of plotting objects to matrix

As shown in the code segment above, we first generate a 640 x 640 matrix, which purpose is to place the desired bounding box generate by the coordinates in previous .json file. In the for loop, the code will process a Metrix for each frame, and create a unique dictionary for each type of object detected in Yolov5. By using the coordinates, we now have the object shown in the matrix, so that we are allowed to do the prediction based on the different libraries and tools including cross validation and convolutional neural network. In addition, for purpose of eliminating minor differences within certain frames, the code is also design a algorithm, which can automatically shift, flip, and rescale the bounding box, if it gains a high confidence level of matching object.

4. EXPERIMENTS

This paper conducts two sets of experiments for a series of model solutions that we aim to combine into one approach. This includes testing both the overall accuracy of a selection of models as well as the average speed for a selection of models, for each type of solution. The first experiment ties directly to model accuracy as it is of paramount importance.

4.1. Evaluation of the Searching Effectiveness

In the case of audio classification, each experiment tests with different levels of data, one of the experiments has two elements undergoing cross validation with five splits, while the other experiment receives five elements and the same amount of cross validation splits. Both experiments point out that the Support Vector Machine (SVC) model has the highest accuracy with 76% and 53.06% respectively. Meanwhile, the testing dataset is created by using three seconds slices of a single movie.

For the object detection solution, under the framework of cross validation with doubled data received the highest accuracy of 43.04%, which is achieved under SVC model, however the model emphasizes an outstanding result in terms of effectiveness, the timing of SVC model gives a hint that this is still debatable or considered as an efficient choice.

The most outstanding outbreak achieved in this research is that of the highest accuracy in our image classification method. By utilizing neural networks and several libraries in TensorFlow. Our neural network image classification achieved a 90% when tested against its training dataset while achieving a 71% accuracy in the validation dataset testing. As shown below in right-hand side of the figure 6, the correlation between self-prediction regarding the validation and testing slope becomes more and more readable, more importantly, the slope reaches a stable level as it approaches the end. By increasing the epoch will not highly affect the accuracy in this case, and the potential overfitting issues has been addressed by implementing dropout layer to neural network and data augment will essentially get rid of similar screenshots. One the left-hand side the figure 6 emphasizes the slope correlation before eliminating the over fitting issue.

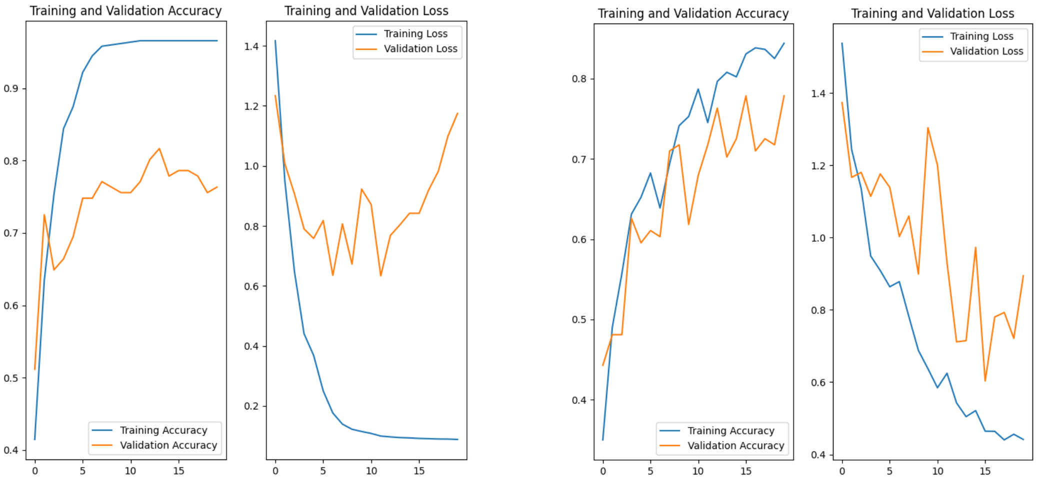


Figure 6. Overfitting model vs. Model with drop out layer

The parallel programming’s accuracy is a mean average of all other three approaches, which are approximately at 55.94% accuracy. In fact, the program utilizes a different approach than picking the most votes from three approaches. Since the image classification reaches the highest breakthrough, the program will take object recognition and audio classification into consideration only if in the situation where neural network image classification ends up with a confidence level under 60%.

4.2. The Evaluation of Searching Efficiency

Every program needs to consistently consider the tradeoffs between efficiency and effectiveness. The result for the audio classifier model indicates that although SVC has the most satisfying accuracy, however, in terms of speed, the Gaussian I Bayes (GaussianNB) gives relatively lower accuracy with 47.34% of accuracy and seventy-two times faster than SVC. As shown in the figure 7, the comparison is made under the unit of seconds.

Likewise, the object detection emphasizes a similar outcome to that of audio classification models. The SVC model once again gives the highest accuracy, as described above, and the relatively slowest time when compared to other tested models. The GaussianNB has a lower accuracy of 42.67% but finishes the task ten times faster than SVC.

Image classification under a neural network reaches the highest accuracy across all of our approaches, in terms of efficiency, it is actually considerable given that it takes 0.103 seconds to make the prediction.

The average timing for parallel processing varies on how many steps are involved. If the program’s image classification reaches a confidence level higher than 60%, the average for prediction is how long it takes to do the image classification. In case the confidence level drops under 60%, two other approaches will get involved as a backup resource to support the result, and it will basically take approximately a total of 2.25 seconds.

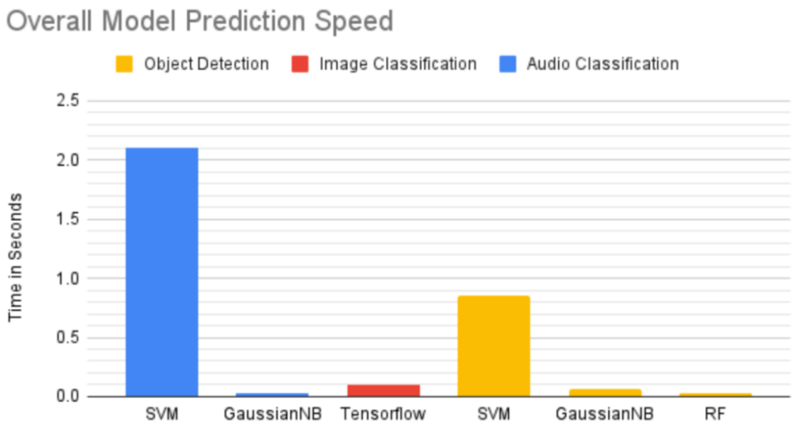


Figure 7. Subsample of machine learning speed tests for each method

4.3. Further Experiment Result

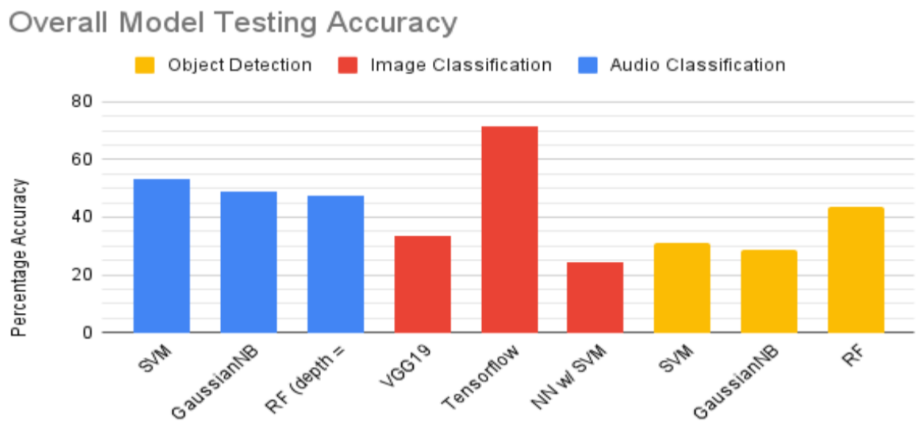


Figure 8. Subsample of machine learning accuracy tests for each method

To select a model, we compare various models and then run multiple tests across them such as cross-validation to pick the most accurate one. We then do hyperparameter tuning to tune it to exactly what we need. For Image Classification we compared SVM, RandomForest, passive aggressive, and convolutional neural networks with results of 65%, 77%, 28%, and 91% respectively.

For object detection, we run the prediction under different numbers of dataset. When cross validation at level 2 with no data duplication, we compared SVM, GaussianNB, RandomForest (max depth 2), RandomForest (no max depth) with result 31.4%, 25.26%, 31.86%, 33.33% respectively. When cross validation at level 7 with no data duplication, we compared SVM, GaussianNB, RandomForest (max depth 2), RandomForest (no max depth) with result 31.1%, 28.57%, 32.23%, 43.25% respectively. When cross validation at level 7 with data duplication, we compared SVM, GaussianNB, RandomForest (max depth 2), with result 43.04.%, 42.67%, 37.17% respectively.

In case of audio classification, we compared cross validation at level 5 with 5 elements. The following models has been tested, SVM Classifier (MidTerm Features), Random Forest (MidTerm Features, max_depth = 2), Random Forest (MidTerm Features, no max_depth), Passive Aggressive (MidTerm Features), GaussianNB (MidTerm Features) with result 53.06%, 48.97%, 46.53%, 37.96%, 47.34% respectively.

5. Related Work

X. Gong and F. Wang [7] noticed that the result of tennis video detection will be significantly more efficient by applying the audio detection. This audio-based approach assists the detection process with a video including a tennis event. With a shortcoming that the support vector machine supports a maximum of two classification problems at the single time, the author implements a method of combination of support vector machine and decision tree support vector machine to tackle the audio multi classification problem. L. Wang, H. Liu, and F. Sun [8] implemented a soft coding bag of dynamic systems to perform a satisfactory performance in the area of extreme learning machines. The paper emphasizes the system efficiency when it applies to the public database with algorithm and comparison in between the industry common method and this BoS approach. The main difference between this work is that we are using different dataset. V. Lopez-Vazquez., et al. [9] emphasizes that classic deep learning approaches are able to tackle the complex neural networks in the area of unexplored environments with even low-quality images. Moreover, the author indicates that with a higher rate of enhancement image could potentially assist the work of detection of features and gain better classification accuracy. In addition, deep learning approaches have been used to detect underwater animals in this paper. J. Gao., et al. [10] implemented several approaches including temporal boundary regression, temporal unit regression network, and more to address current problem in long untrimmed videos with generate temporal actional proposals.

6. CONCLUSIONS

Three approaches are capable of delivering a valid result with high confidence analysis for the video clip prediction. By utilizing the parallel programming, the final state of application by combining three approaches will finalize and leverage the result both in terms of effectiveness and efficiency. The accuracy level of image classification emphasizes that in ordinary situations, the prediction is strong enough to provide a meaningful result. The application requires a comprehensive dataset in order to make a prediction at a high level. By asking the user to upload their video clips and movies will enhance the database and the categorized dataset will save the result for future references, which means that while the user uses it, the application is learning and analyzing at the same time. All of the classification approaches present at least 90% accuracy when the database analyzes it in advance, even if there are minor differences when capturing the frame and voice, since the application has been designed to handle it by rescaling, rotating, shifting the frame. Even with the video clip that the database has never seen before, the image classification will brief the prediction and confidence level and the application will decide whether or not to involve the other two approaches. The prediction for the validation dataset is proven to be trusted and accurate. Speaking of image classification, in the future, the application will not limit to the current defined architecture and libraries, and push the accuracy to state-of-art performance by implementing the application under other existing techniques such as PyTorch. Likewise, the performance of object detection and audio classification will also rebuild to advance the result.

In the application, all of the approach is still not applying state-of-art performance models, which means that once it is getting published, this application has already become a past tense. The growth rate of deep learning technologies just requires tons of research readings and learning from the other’s work. The limitation of human resources causes the application has not tried all of the existing methods. On the other hand, mobile devices are the most popularly used device currently and sadly we have not decided when to release the mobile application due to the limitation of human resources to maintain the system in the future.

In the future release, several models will be given consideration and implementation for a single approach, as we want to select from new models that are developed and could potentially increase accuracy, but also back up the experiment with stronger evidence. Also, enhancing the user experience on a mobile devices’ browser will be strongly considered.

REFERENCES

[1] S. Moral-García, C. J. Mantas, J. G. Castellano, M. D. Benítez, and J. Abellán, “Bagging of credal decision trees for imprecise classification,” *Expert Systems with Applications*, vol. 141, p. 112944, 2020.

[2] F. Alam, R. Mehmood, and I. Katib, “Comparison of decision trees and Deep Learning for object classification in autonomous driving,” *Smart Infrastructure and Applications*, pp. 135–158, 2019.

[3] M. A. Friedl, C. E. Brodley, and A. H. Strahler, "Maximizing land cover classification accuracies produced by decision trees at Continental to Global Scales," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 969–977, 1999.

[4] X. Yang, X. liu, M. Jian, X. Gao, and M. Wang, "Weakly-supervised video object grounding by exploring spatio-temporal contexts," *Proceedings of t^{he} 28th ACM International Conference on Multimedia*, 2020.

[5] J. Ye, X. Lin, L. He, D. Li, and Q. Chen, "One-stage visual grounding via semantic-aware feature filter," *Proceedings of t^{he} 29th ACM International Conference on Multimedia*, 2021.

[6] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "Learning visual affordance grounding from demonstration videos," *arXiv.org*, 12-Aug-2021. [Online]. Available: https://arxiv.org/abs/2108.05675. [Accessed: 23-Dec-2021].

[7] X. Gong and F. Wang, "Classification of tennis video types based on machine learning technology," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–11, 2021.

[8] L. Wang, H. Liu, and F. Sun, "Dynamic texture video classification using Extreme Learning Machine," *Neurocomputing*, vol. 174, pp. 278–285, 2016.

[9] V. Lopez-Vazquez, J. M. Lopez-Guede, S. Marini, E. Fanelli, E. Johnsen, and J. Aguzzi, "Video image enhancement and machine learning pipeline for underwater animal detection and classification at cabled observatories," *Sensors*, vol. 20, no. 3, p. 726, 2020.

[10] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

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

[12] Brezeale, D., & Cook, D. J. (2008). Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(3), 416-430.

[13] Arthurs, J., Drakopoulou, S., & Gandini, A. (2018). Researching youtube. *Convergence*, *24*(1), 3-15.

[14] McDonald, K., & Smith-Rowsey, D. (Eds.). (2016). The Netflix effect: Technology and entertainment in the 21st century. Bloomsbury Publishing USA.

[15] Yan, L. C., Yoshua, B., & Geoffrey, H. (2015). Deep learning. nature, 521(7553), 436-444.