University of Melbourne

COMP90055 Computing Project(25 Credits)

A Mobile Application for Poisonous Spider Detection based on Deep Learning

Kangyun Dou 740145 Site Huang 908282

School of Engineering

Type of Project: Software Development Project

Supervised by

Prof. Richard Sinnott Keating

Summer Semester,2019

25 February 2019

Acknowledgments

We would like to express our deepest appreciation to Prof. Richard Sinnott. He not only provides us the opportunity to do the project, but also give us lots of suggestions and encouragement during this project.
Furthermore, a special gratitude to University of Melborune. Our uni provides us lots of great learning environment and facilities (especially GPUs for this project).
Last but not the least, we owe a "thank you!" to those who provided us the possibility to complete this report.
Thank you all!

Contents

List of Figures

Abstract

Deep learning has recently emerged as an attractive solution for computer vision to gain high-level understanding from digital images and build an intelligent systems to assist humans in various tasks. Moreover, deep learning has been applied broadly in different fields, such as machine translation and handwriting generation.

The purpose of the project is an object detection application software development. A custom detection model based on the concept of transfer learning is built in this project. In addition, some existing detection frameworks are analyzed and compared according to their differences in architecture and prediction effectiveness. Finally, the SSD Mobilenet_v1 FPN is selected as the base model and retrained it for spider recognition.

A friendly IOS application is developed for this project. Google Cloud Platform is used as the medium of exchanging for prediction information. The retrained model is uploaded to the Google Cloud for constructing a machine learning engine. Besides, the prediction accuracy of the retrained model is analyzed and its performance for classifying spiders in different scenarios is verified by IOS application.

Key words: Deep Learning, Tensorflow, Machine Learning, CNN, Posionous Spiders, Image Procressing, Obejct Detection, IOS Application

We certify that

- *this thesis does not incorporate without acknowledgment any material previously*
- *submitted for a degree or diploma in any university; and that to be best of our*
- *knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the test.*
- *where necessary we have received clearance for this research from the University's Ethics Committee and have submitted all required data the Department*
- *the thesis is 5000 words in length (excluding text in images, table, bibliographies and appendices).*

1 Introduction

1.1 Deep Learning

As a subfield of machine learning, deep learning has a broad application potential in speech recognition, image recognition, mobile advertising [1]. Convolutional neural networks become noteworthy ever since the Hinton team participated in the competition of ImageNet recognition and won the championship through constructing a CNN network called AlexNet, which defeated the classification performance of traditional machine learning method of SVM [2].

Fig.1 shows that a traditional machine learning method needs two important components which are a feature extractor and a classification model [3]. However, in practice, due to the complexity of a classification problem and the uncertainty of feature definition, feature extraction becomes extremely challenging. Under such circumstances, it's essential to consider various elements in choosing a well-structured machine learning framework. In contrast, deep learning model can perform feature extraction and classification together, which is effective in lots of complex scenarios. Therefore, compared to traditional machine learning methods, deep learning methods stand out with its robust and significant classification performance.

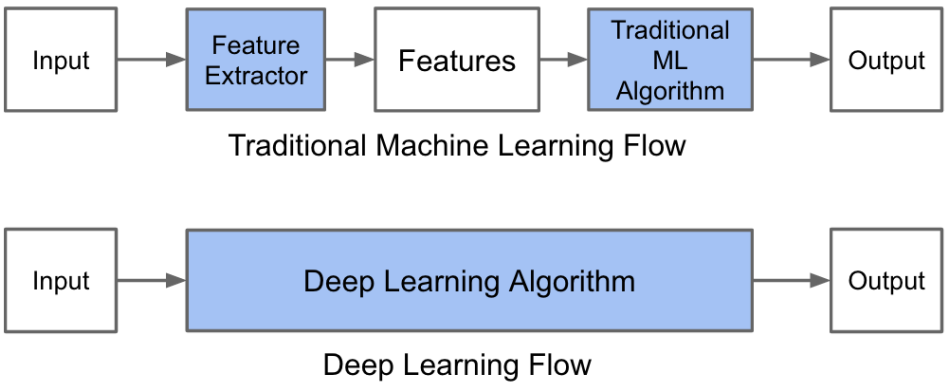


Figure 1: Traditional Machine Learning vs. Deep Learning [4]

Deep learning method performs better than traditional machine learning methods in terms of the processing speed and classification outcomes [5]. The reason is obvious that GPUs are efficient and well-suited to deep learning in processing matrix multiplication and convolution in parallel.

Nowadays, the proliferation of network services is contributing to a dramatic increase in data traffic. Theoretically, the increase of data will improve the performance of classification algorithms. However, Fig.2 shows older algorithms may reach a bottleneck as they cannot learn more from the data and obtain a better payoff. On the contrary, deep learning model with its outstanding feature extraction and well-structured architecture eventually performs better than older algorithms and gains a ever-increasing performance with the increment of data.

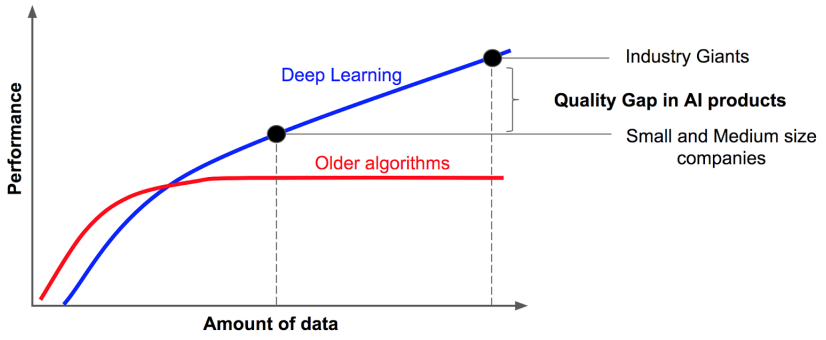


Figure 2: Plot of Amount of Data (x-axis) vs. Performance (y-axis) [6]

1.2 Tensorflow

There are a variety of machine learning frameworks available for deep learning tasks, such as PyTorch, MXNet, Chainer and Tensorflow. Tensorflow developed by Google AI organization is considered as the most well-suited framework in this project.

Tensorflow is an open source library for implementing large-scale machine learning projects. It has an outstanding performance which can handle large mathematical

operations. Moreover, due to its flexibility, the Tensorflow's structure can be utilized to across a variety of platforms (CPUs, GPUs, TPUs), and available in both desktop and mobile devices.

Tensorflow constructed dataflow graphs which contain a series of processing nodes, where each node as well as their connections represent mathematical operations. In addition, Tensorflow provides lots of latest machine learning algorithms and powerful tools, such as Tensorboard, which is used for visualizing the training process and model's architecture.

Therefore, Tensorflow not only fueled the rapid of development behind machine learning projects, but offered additional conveniences for research in building a neural network under a rich and flexible development environment.

In this project, Tensorflow is selected as the deep learning framework to build a custom object detection model and apply it to the mobile application.

1.3 Mobile Application

According to the results of recent surveys, with the proliferation of mobile devices and the ever-growing network services, the number of mobile devices is around 5 billion globally and is expected to be continuously increasing [7]. Mobile devices have changed people's lifestyle for its convenience and portability.

The primal objective of this project is to develop a spider detection application which indicates whether the spider is poisonous or not. The model is trained on high performance computing computers (Spartan is used in this project) and uploaded to the Google Cloud Platform. The design of application is that users can upload an image from their personal mobile device's photo library to the server while the server will use the machine learning engine to predict the spider's breed for that image. Based on that, Google Cloud Platform is used as the server to store user-uploaded images along with their predictions in the database. Google Cloud Platform is convenience and compatible with Tensorflow framework.

A machine learning engine is created for generating predictions based on the re-trained model. Once the predictions are finished, the mobile application will fetch the prediction data from the Google Cloud and suggest whether this spider is poisonous or not. Furthermore, the application design a Wikipedia page view for users to explore more information for spiders.

1.4 Poisonous spiders

It's an interesting topic to discuss how to define "Poisonous" of spiders. Based on recent research on spider species, there are basically two types of venom that have an effect on humans: neurotoxic and cytotoxic venoms. Neurotoxic venoms work directly on the nervous system. The best known example is the venom of the Black Widow/Redback spiders (Latrodectus species). Necrotic venom cause damage to the tissues, such as blisters and lesions. There are no confirmed records of spider bites in Australia causing necrotic lesions, although the bites of Recluse Spiders, which are native to the Americas, have been confirmed to cause tissue necrosis. Generally, neurotoxic venoms kill more quickly than cytotoxic venom.

Even though a few spiders are poisonous, there have been no deaths in Australia from a confirmed spider bite since 1979 because of the effective anti venom [8]. Actually, normal spiders in daily life are not poisonous even though they may look horrible. However, there are deaths recorded because of spider shock, which happens when a person spotted a spider surprisingly and then accidentally get hurt by other incidents (such as fell over the head). To prevent these tragedies happened, the purpose of the project is to make people know about the spiders and treat them fairly.

Therefore, based on the types and poisonous levels, in this project, 3 categories and its corresponding advice are defined as :

(i) Deadly! Get rid of it ASAP!

(ii) Poisonous. Bites may be painful. Be careful!

(iii) Low Risk. They are beneficial in control of flies.

2 Dataset

The dataset is crucial to this object detection project. There are three key information needed, which are clear label classification, high-resolution images, and bounding box information. For this poisonous spider detection project, there is no existing spider dataset available for us. Therefore, the dataset for better detection outcome is gathered from online.

2.1 Crawling Dataset

There are mainly three major sources for the dataset since there is no existing dataset available for the project, which are Bing, Google, and Flickr. Regarding the crawling, there are three difficulties involving spider families determination, dataset accuracy, and limited data.

For the spider families, there is no common agreement on the accurate type [9]. In 2015, taxonomists suggested the number of spider species could be around 45,700 and total of 113 families [10]. However, the most common spider families division is around 20 different classification categories since 1900 [11]. In this project, from the aspect of the genus, 17 types have been selected based on the Australian Museum's research on Australia spider type [12].

The second problem is that the quality of the dataset needs to be controlled. For the following figures below, it's obvious to notice that the female spider and the male spider looks different in many aspects. Therefore, it would impact the outcome of prediction enormously.

The third problem is caused by the limitation of dataset volume. The spider is a minor field, so there are not sufficient high resolution pictures in the internet compared to other common creatures such as dog or cat. For the above difficulties,

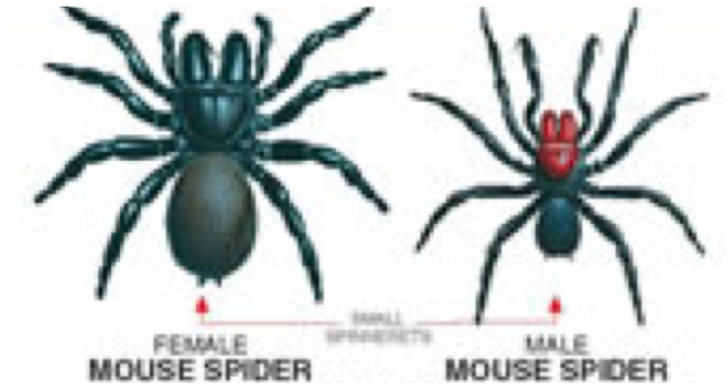


Figure 3: Mouse Spider: Female vs. Male

not only the dataset needs to be purified as well as the direction of the pictures to be changed to better train the model.

When crawling data, both the spider's scientific name and nickname are used as keywords for searching. If the keyword does not contain "spider", the keyword will be added to the searching keyword text.

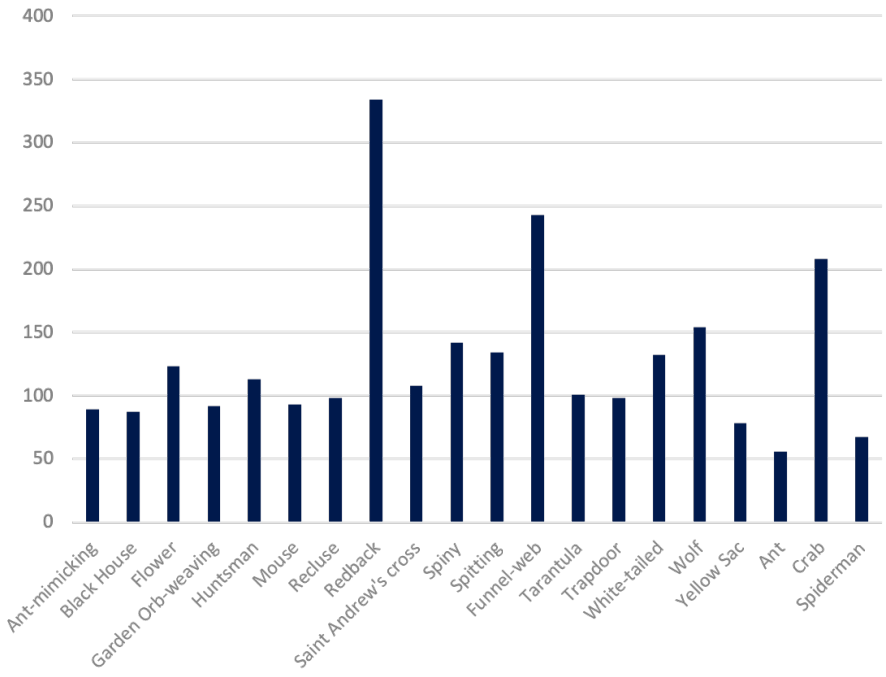


Figure 4: Crawling Dataset Original Amount

After crawling, there are total of 17 types(Ant-mimicking Black House, Flower, Garden Orb-weaving, Huntsman, Mouse, Recluse, Redback, Saint Andrew's cross Spiny, Spitting, Funnel-web, Tarantula, Trapdoor, White-tailed, Wolf, Yellow Sac) and three other common confusing types (crab, ant, and spiderman). The exact amount for each species is listed in the Fig.4.

2.2 Purifying the dataset

The crawled dataset could not meet the expectation because it may contain lots of useless information. For example, when the key word is "flower spider", many related flower pictures will be crawled, which are useless for this project. Therefore, those irrelevant pictures need to be deleted from the dataset.

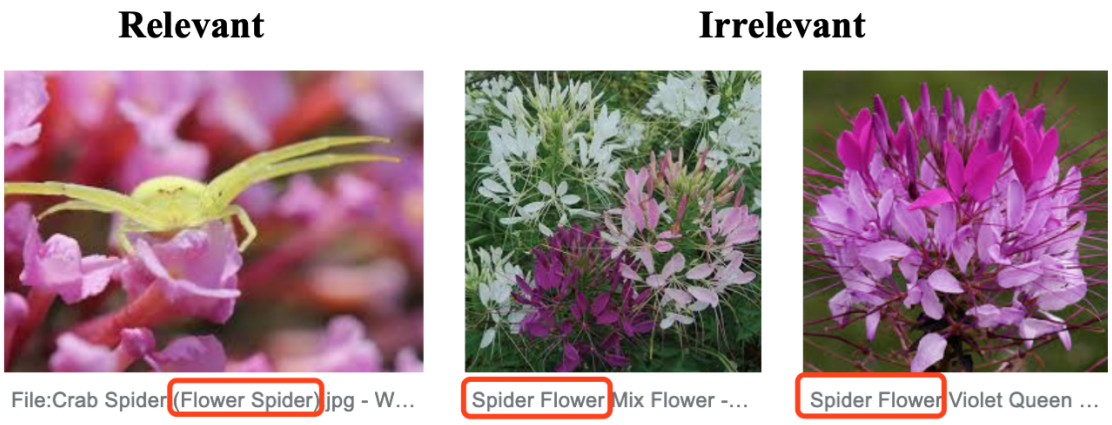


Figure 5: Some of the flower spider result

After purifying, the amount of dataset decreased, which increase the difficulty of prediction. And the detailed amount is illustrated by a bar chart in Fig.6.

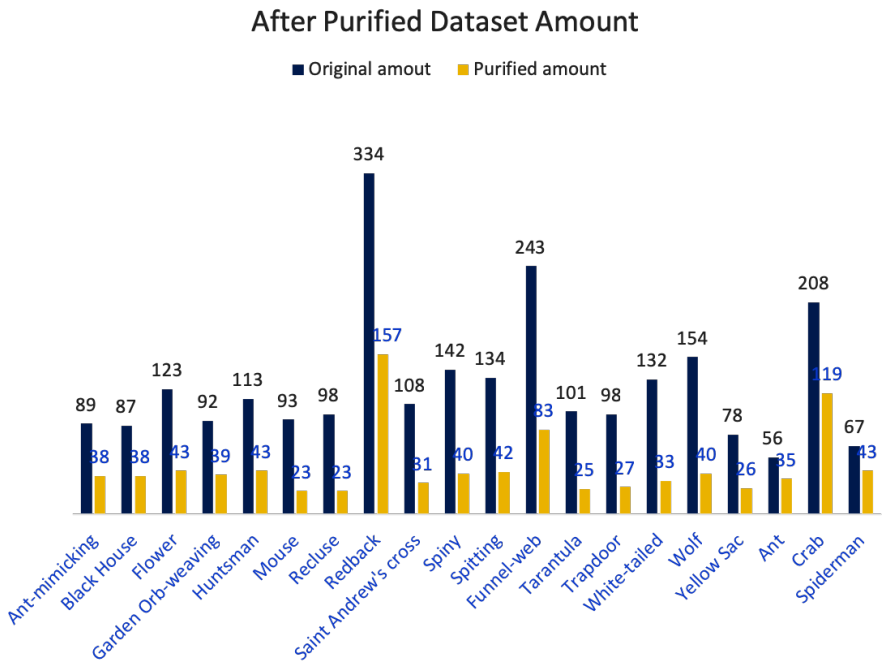


Figure 6: Data amount after purity

3 Image Preprocess

3.1 Data Augmentation

Due to the limited resources on the internet, the dataset for the project is far below the expectations and requirements. Therefore, data augmentation is used to expand the dataset by applying transformation and deformations on the images. From previous researches, cropping, rotation scaling, and transformation are often used for generating more data [13]. In this project, rotation is used to obtain more data from various perspectives. The reasons is obvious that the spider's side face may look completely different from each other. Convolutional neural network models can learn more from different aspects from one picture.

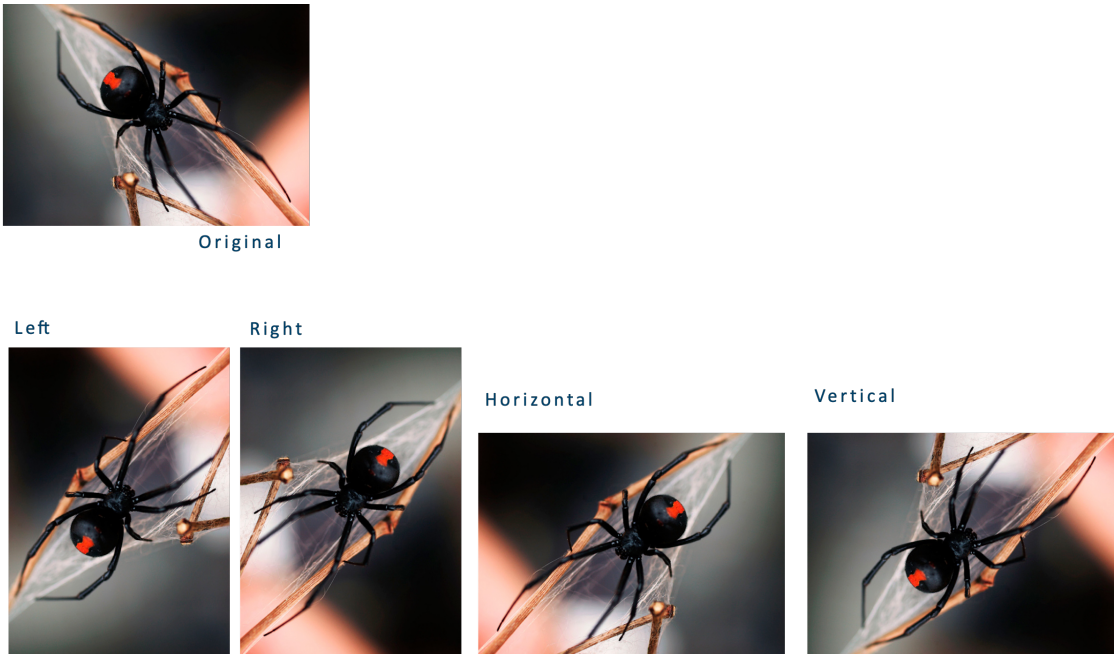


Figure 7: Examples of rotation

3.2 Resize

It's important to realize that resizing a picture is can be a double-edged sword. Higher resolution images for the same model have better classification accuracy but slower to process. Therefore, on the positive side, resizing can speed up the training process. On the negative side, it could lead to lower classification accuracy and result in bad prediction. Resizing all images so as to speed up the training process is preferred in this project.

After done the image preprocess, it's time to draw the bounding box for each image to indicate the exact location of an object. Fig.8 shows the procedure of image notation which is labeling an object in the picture and then generating XML file contained the bounding box information.

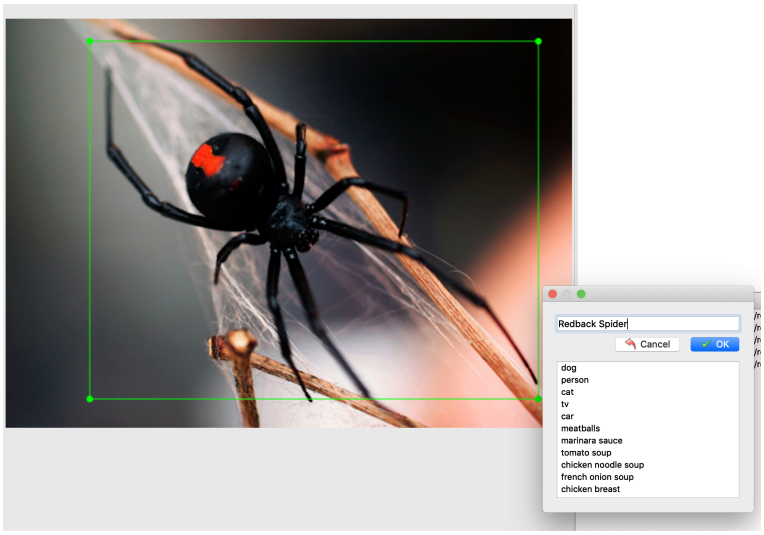


Figure 8: Label Image with bounding box

3.3 Mean Normalization

Photoshop is a frequent tool for photographers to make the picture more appealing. For instance, the below image which contains a brown spider (funnel web spider) with brown background is easy for human eyes to recognize while it is difficult for computers to figure out. Therefore, the mean normalization method is used to make computer better differentiate spiders from the vague background.

Figure 9: Spider in similar color background

Mean normalization, a tool which focuses on the comparative color of the image instead of the exact color (RGB values), is used to improve the accuracy and enhance the performance of the training model.

3.4 Shuffle

Shuffling is another key point in optimizing the model before generating TF record file recording all the information of dataset. By reading from them using a command tf.train.string_input_producer with shuffle=True, the order of input data files will be randomly shuffled. TF record binary data files which have characteristics of fast reading speed and less disk space are often used in machine learning projects.

To sum up, all the dataset information after pre-processing is illustrated as the figure below. Fig.10 shows that even though the data can be gathered from diverse website sources, in practice, only a small amount of data can be used in the project. The reason is obvious that without purifying and processing the dataset, it could lead to a irreversible impact on prediction accuracy of the retrained model.

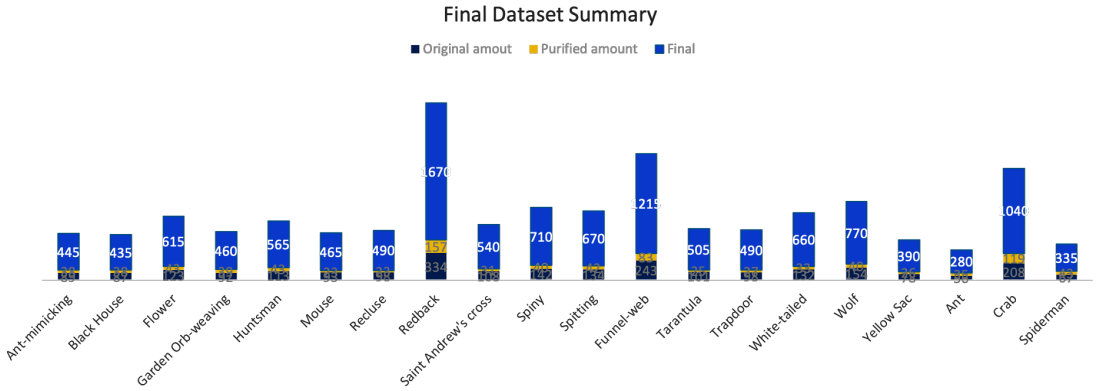


Figure 10: Final Dataset Amount

4 Applying Deep Learning

In this section, the basic concept of transfer learning is introduced and discuss different object detection models architectures, as well as their main differences is discussed and investigated.

4.1 Transfer Learning

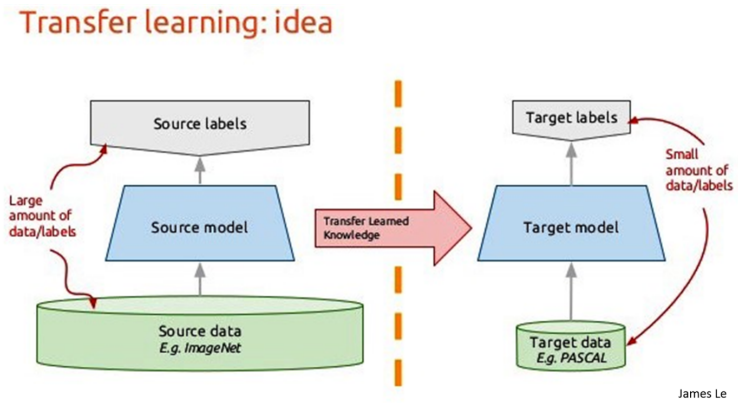


Figure 11: Transfer Learning Process

Transfer learning is a theoretical approach applied to train a custom model based on the pre-trained models. In that case, a custom object detection framework can be built based on the existing pre-trained models instead of random initializing parameters and building a network [14].

From the research based on goal-oriented chatbot dialog management bootstrapping [15], the authors compared differences of models with and without transfer learning. As Fig.12 shown, the model with transfer learning gains a better performance and has a higher success rate than the model without. Transfer learning is effective and efficient because it takes the weights and parameters of a pre-trained neural network and fine-tuning the model with the custom dataset. Transfer learning works mainly because it supports to replace the last layer of the network with

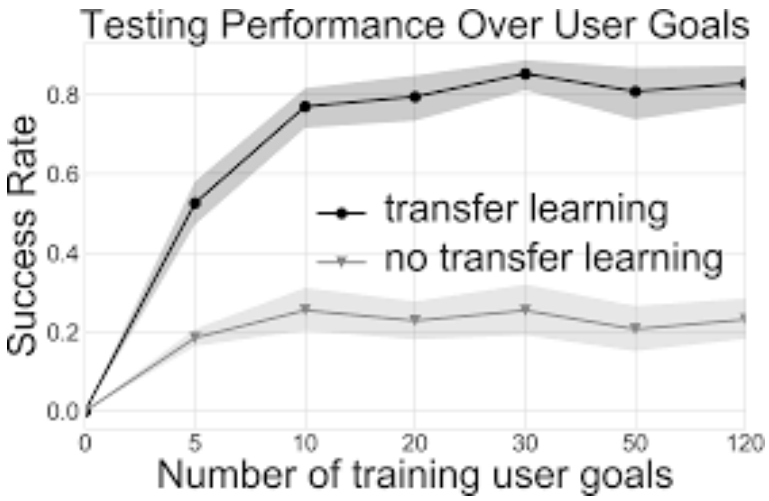


Figure 12: Transfer Learning Benefits [15]

the custom defined layer (For Spider Recognition), and freeze the weights and parameters of the other layers.

Therefore, a custom detection model can be built with a limited amount of training data through transfer learning approach, while benefiting from additional information from the source already learned from previously trained models.

4.2 Object detection framework

Nowadays, lots of promising solutions are proposed for object detection tasks [16]. The authors in [17] proposed a novel method using deep neural network to predict multiple groundtruth locations in an image. A multi-scale convolutional MultiBox approach is proposed in [18] for high-quality object detection, which significantly improves the proposal quality and thus enhances the overall performance in detecting objects.

4.2.1 Faster R-CNN

Faster R-CNN is an improved version of Fast R-CNN. Their main difference is Faster R-CNN uses "Regional Proposal Network (RPN)" whereas Fast R-CNN

uses selective search for generating region proposals.

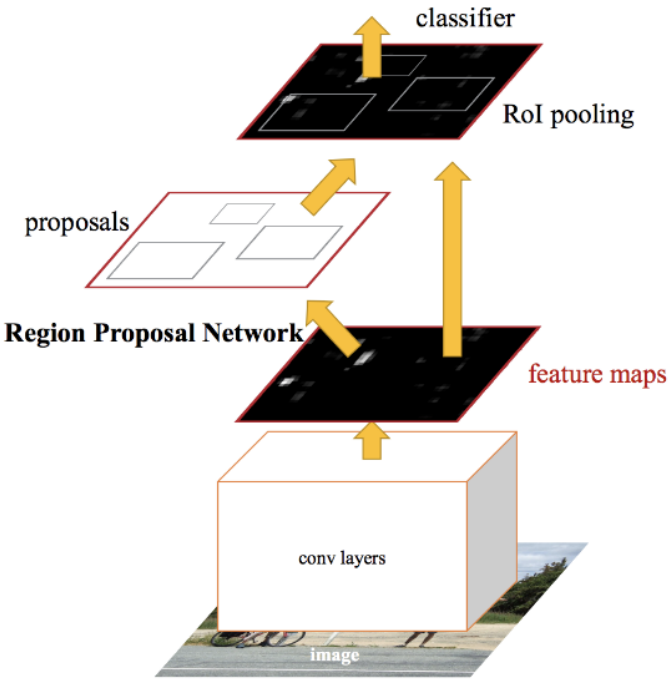


Figure 13: Faster R-CNN [19]

Firstly, the image is taken as input to the convolutional network which returns a feature map. And then the regional proposal network takes feature maps as input and generates a series of proposals along with their objectness score [19]. During the process of generating region proposals, anchor boxes with different aspect ratios are applied to each location of that image. Finally, object proposals are passed to a fully connected layer to generate output predictions and bounding boxes for objects. RPN is the essence of Faster R-CNN, which has outstanding detection accuracy. However, compared to the SSD model, it takes relatively longer computational time during the process of detecting objects [19].

4.2.2 SSD

SSD model is based on a feed-forward convolutional network that generates predictions containing a series of bounding boxes as well as confidences of the object

[20]. SSD model has a higher detection speed compared to Faster R-CNN model, but Faster R-CNN is more precise than the SSD model.

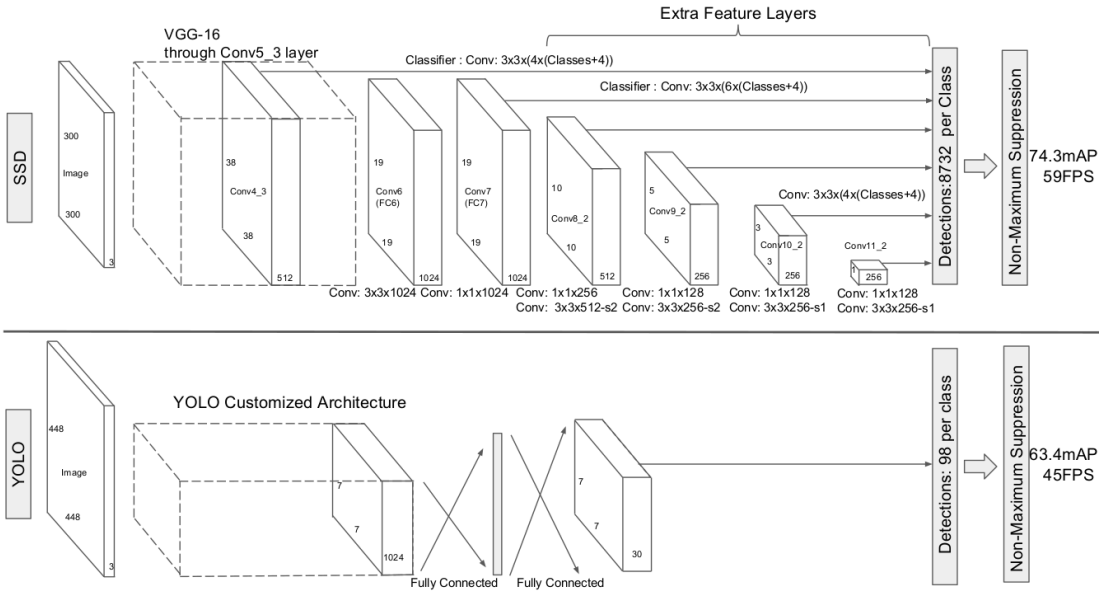


Figure 14: SSD vs YOLO [20]

SSD model adopted the anchor mechanism of Faster R-CNN. However, the anchors produced is a little bit different from Faster R-CNN which adjust each location in the image. The way of generating anchors by the SSD model is more similar to YOLO.

The main difference between SSD and YOLO is that SSD has more feature layers at the end of the network, which is used for predicting the offset to original boxes of different aspect ratios [20].

4.3 SSD Mobilenet_v1 FPN framework

In this section, the basic structure is discussed and the potential application of MobileNet and FPN in object detection is investigated. Besides, the primal objective of this project is to build an application detecting spider species. In order to ensure a satisfied prediction accuracy and a reasonable detection speed, SSD Mobilenet_v1 FPN is chosen as the object detection model for this project.

4.3.1 MobileNet

The authors in [21] proposed a network architecture called MobileNet. It is based on depthwise separable convolutions, which using width multiplier and resolution multiplier to balance the prediction accuracy and network latency so as to obtain a better payoff. Streamlined architecture based MobileNet has a satisfied predicting performance for embedded mobile application [21].

4.3.2 FPN

It's a fundamental challenging task in computer vision to recognize an image with different scales. A promising solution is the FPN (Feature Pyramid Network) comprising a bottom-up and a top-down pathway. The bottom-up pathway is a process of feature extraction to build a feature hierarchy [22]. The semantic value of each layer is increased as the high-level feature hierarchy is constructed. SSD model only selects the upper layer for object detection due to the highest semantic value. However, the resolution in the upper layer is not high enough to recognize the complete features of an image [22]. Therefore, the SSD model doesn't have a satisfied performance in detecting small objects.

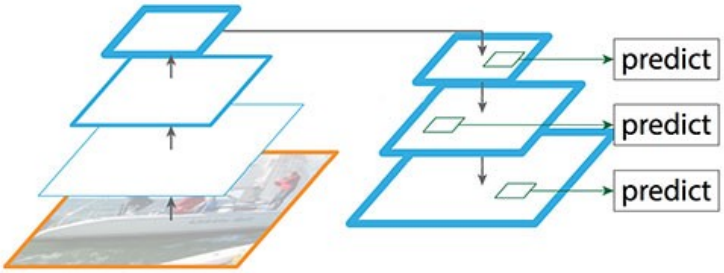


Figure 15: FPN

As Fig.15 shows, FPN follows the top-down pathway to build high-level semantic feature maps for each semantic rich layer. A lateral connection is added at each feature hierarchy structure to better detect the location of object in an image [22].

5 Results and Analysis

5.1 Evaluation Metrics

For testing dataset, 50 pictures of each class are picked out randomly to do the model evaluation. Two evaluation metrics, accuracy and confidence, is introduced in this project to better evaluate the performance of the model.
In machine learning concept, Confidence defines the probability of the event (or probability of input to fall in different classes). If a class has a high probability of this particular class, then it has high confidence.

On the other hand, accuracy defines the ability of the learning algorithm to predict accurately. It defines the percentage of correct predictions made from all predictions.

5.2 Prediction accuracy of each class

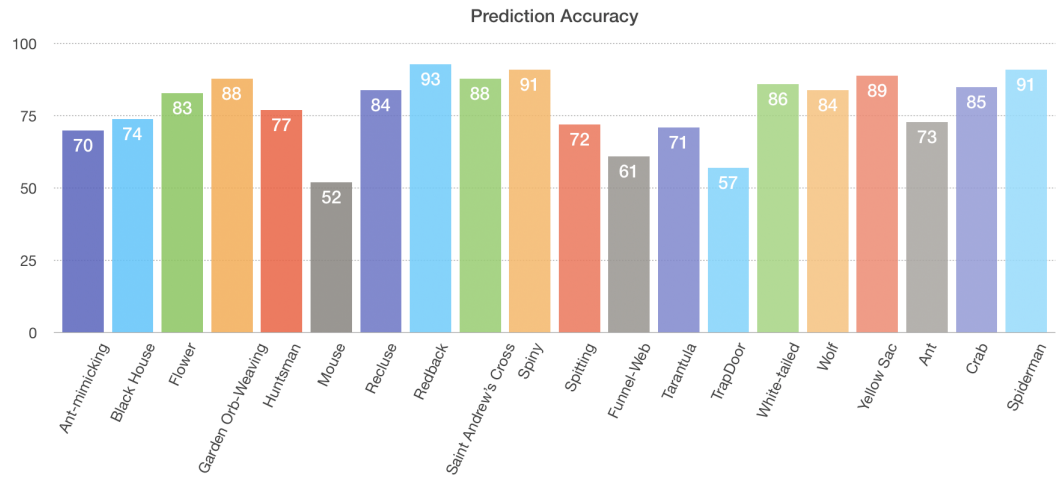


Figure 16: Prediction Accuracy

Fig.16 shows the accuracy of each class. The reasons why different types of spiders lead to differences in accuracy are investigated as:

- Distinct vs. Similar appearance:
  In general, spiders with distinct characteristic is easy to recognize, such as redback spider (with red color on the back) and spiny spider (have spun on the back). Whereas there is no a significant difference among mouse spider, funnel-web spider and trapdoor spider in shapes and colors. Therefore, the model achieves relatively higher accuracy in recognizing redback spider and spiny spider. Especially, the accuracy is rather low among mouse spider and trapdoor spider, which is around 50%. This is because they look very similar on the appearance. Normal people could not be able to detect them directly by eyes.

- Shape difference:
  It is interesting to notice that the model could detect spiderman with 91% accuracy. Even though the spiderman has common with the red back spider with mostly black and red in color. However, the shape of human and spider are totally different. In that case, the model performs well prediction on differences of shapes.

- Dataset amount:
  In General, dataset amount positively affects the quality of prediction. The limited dataset could cause inaccurate prediction even though a well-structured model. Therefore, there is no surprise to find that the species with limited available dataset performs worse prediction in this case.

- Different photograph angle:
  The most spiders' images among the dataset are taken from the upper angle. In addition, different image angles may cause differences in prediction because some types of spiders have their more or less distinct characteristics for people to recognize. Without the complete and clear characteristics of a spider from an image, it may result in low accuracy of prediction.

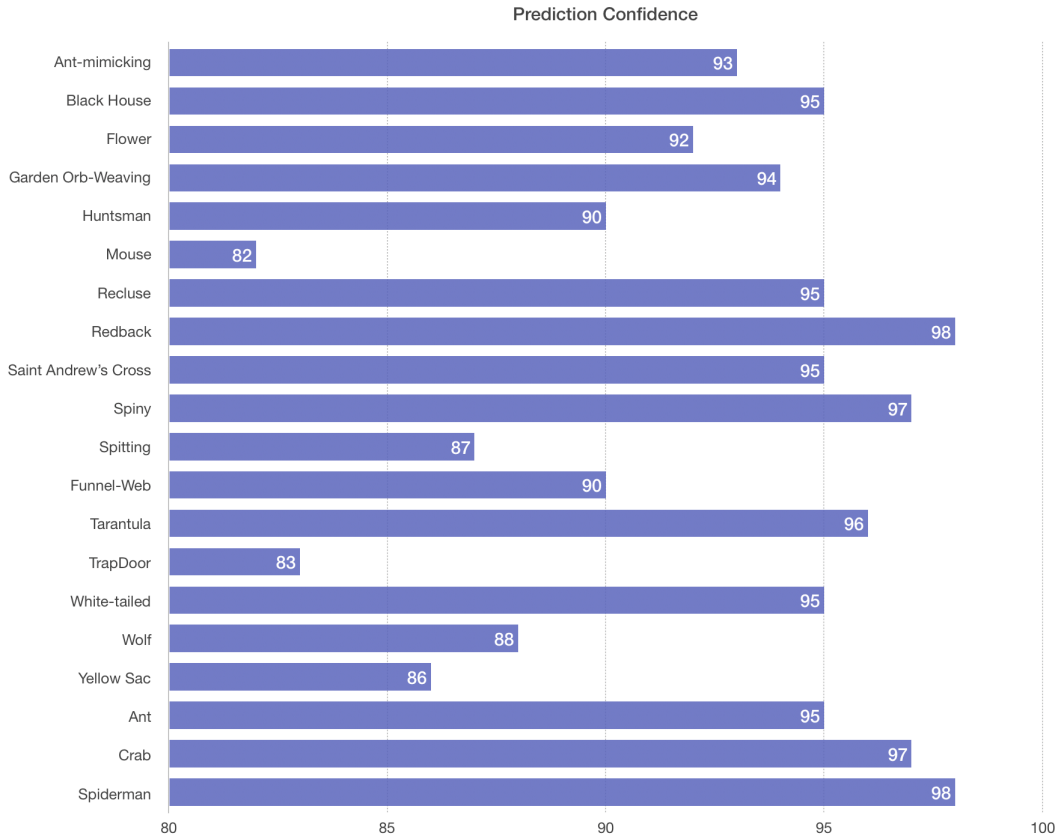


Figure 17: Prediction Confidence

5.3 Prediction confidence of each class

Fig.17 show how confidence on average the model will predict for each class. Generally, the higher the prediction accuracy, the higher confidence the model will be.

5.4 Training Analysis

5.4.1 Learning Rate

The learning rate curve rises rapidly at the first 2000 steps to quickly adjust the model fine-tuning the new dataset, and then decreases slowly and reach nearly 0

when it has been trained 23,500 steps, which means that the model now already converges to the optimal point.

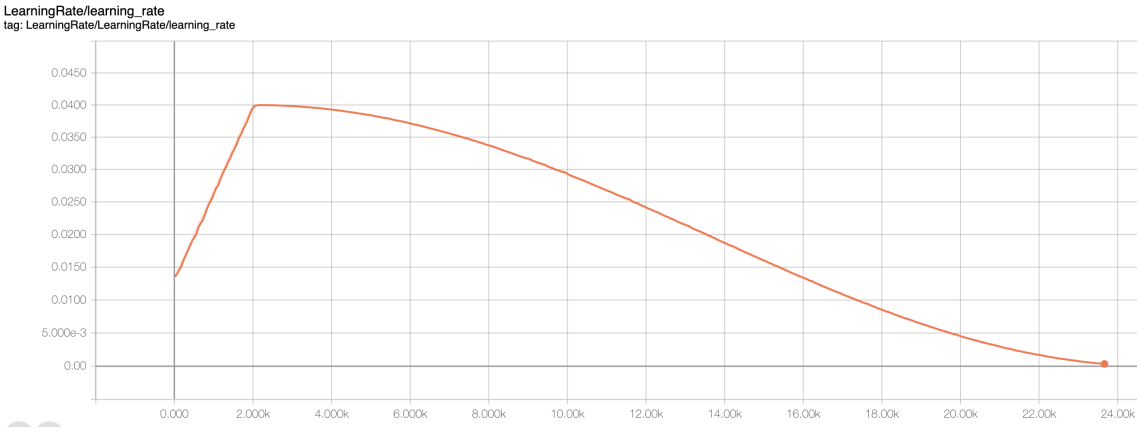


Figure 18: Learning Rate

5.4.2 Total loss

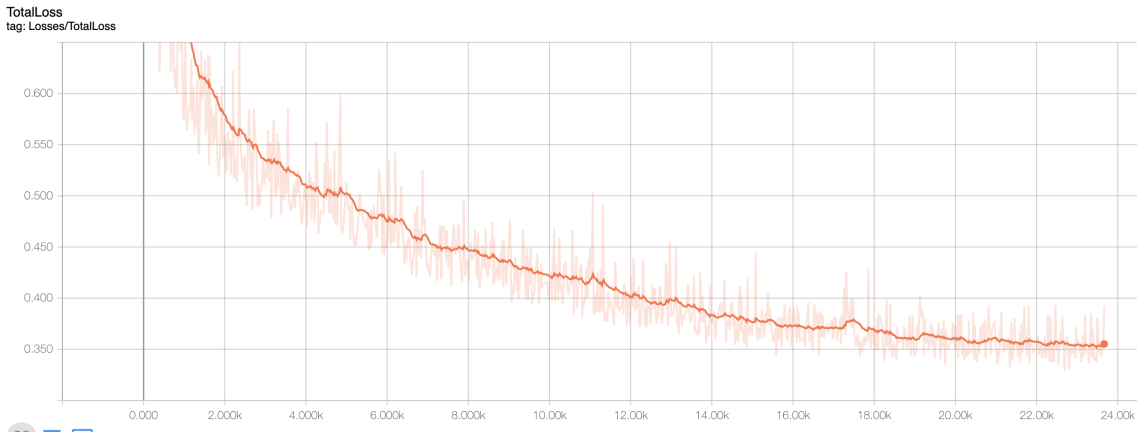


Figure 19: Total Loss

According to the learning rate curve, the slope of loss curve as shown in Fig.19 is also steep at the beginning 2000 steps due to the impact of self-adjusting the model. Then the curve decreases slowly to make the model to find the global optimum point. Finally, the loss curve reaches stable when the model has trained

around 23,500 steps. That is to say, the model has reached the global optimum and ready for spider detection with excellent classification performance.

5.5 Detection scenarios

Form the perspective of user's application scenarios, there will be one separate spider detection and maybe three spiders detection.

5.5.1 Similar Spiders

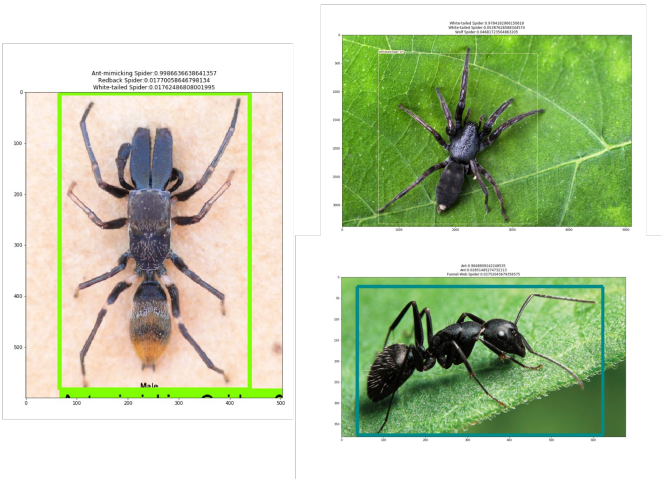


Figure 20: Similar Spiders Outcome

Two types of spiders are chosen to make comparisons with the ants because these three creatures look similar on the appearance. White-tailed spiders, as the name indicates, have a white tail, which is a very unique characteristic. Another specie is called ant-mimicking spider, the spiders in this kind are different individually. They are probably different from color, but they are special in hand. The shape of the ant-mimicking spider is like a pillar. The SSD model can learn from these slightly distinct characteristics. From the result, these three species are identified by more than 95% of accuracy. Therefore, as long as the picture contains complete characteristics of a spider, there is a high probability that the model could recognize the species accurately.

5.5.2 Two Spiders in one image

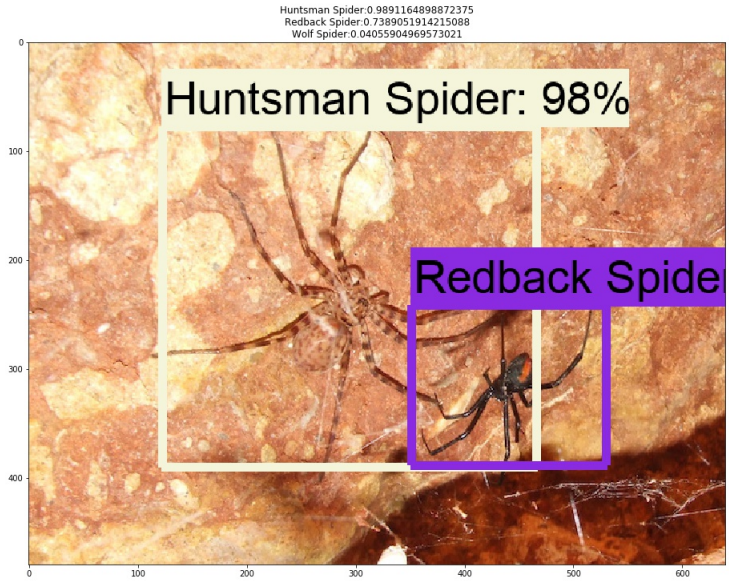


Figure 21: Two spiders in one image

The scenario of two spiders in one image is also tested by the model. From this example, the huntsman spider and redback spider can be detected without mistakes. However, the model is more confident to predict the huntsman than the red back. That's because the redback spider is smaller than the huntsman spider shown in the picture. Moreover, the prediction accuracy of the SSD model is sensitive to the object size in the image. Therefore, the essence of an accurate prediction is to ensure the object clearly showing in the image.

6 IOS Application

In this section, the implementation details of the IOS application is introduced. A basic front end IOS application is built for detecting spiders. Google Cloud and Firebase are used to build a server-client model for the project.

It's effective and efficient to use Google Cloud Platform as the server. Firstly, this saves the storage of the client application. The client end is a thin application which its function is just uploading an image and fetch data from remote. Secondly, the server performs more stable than the mobile application. The performance of the client-only application relies highly on the mobile device's hardware level. However, the drawback is obviously restricted to the network bandwidth.

6.1 Google Cloud Machine Learning Engine

Google cloud provides plenty of opportunities for developers to build machine learning models with multiple ML frameworks, including scikit-learn, XGBoost, Keras, and TensorFlow [23]. The online prediction deploys ML models with serverless, fully managed to host that responds in real time with high availability.

6.2 Firebase

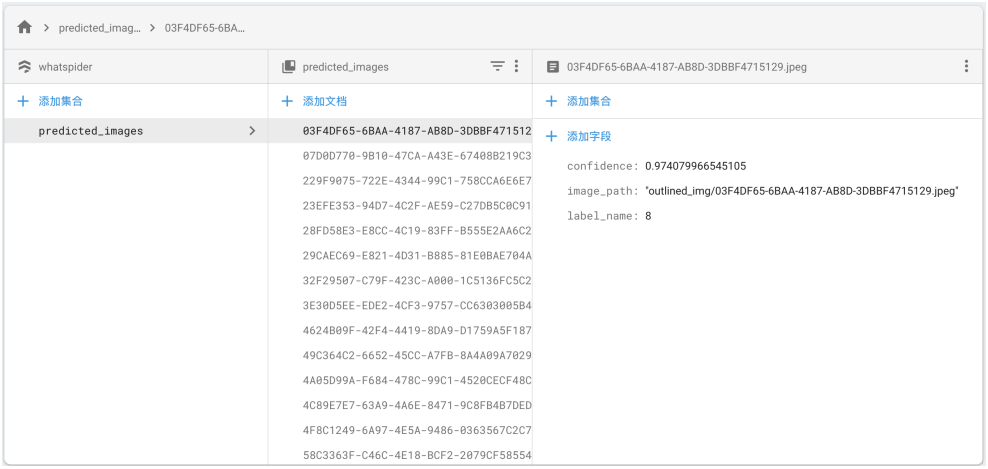


Figure 22: Cloud Database Overview

In the project, Google Cloud is worked as cloud storage to save images. The users upload an image for recognition to the specific Google cloud storage. And then the outlined images generated by the machine learning engine are saved in the

storage. Additionally, the cloud database will also save the outlined image path and the corresponding prediction confidence at the same time.

Furthermore, cloud function is also used for connecting machine learning Engine with Firebase storage. It will be triggered whenever a new image added in that specific cloud storage bucket. Then the cloud function will run prediction for that new image through the machine learning Engine. Finally, once the response from machine learning engine is received, cloud function will be triggered to save all the information of the new outlined image.

Meanwhile, the client app will fetch prediction data from the cloud, so that the users could get the prediction confidence and the prediction name, and download the new image with the bounding box around according to the image path. In this way, the client app does not need to keep pulling the data from the server. It will get updated when the prediction is finished. So with the help of Google cloud, it leverages the existing technologies that are machine learning Engine, cloud storage and cloud function, which is very handy and effective.

6.3 Application of Firebase and Google Cloud

To begin with, the model is uploaded to the Google cloud. And Google cloud will run the trained model and then return a list of bounding boxes that indicate the location of the object in an image. In this step, Google cloud machine learning engine API is used to deploy the model. One of the benefits using Google Cloud and Firebase is serverless. All the client doing is just uploading an image to the specific firebase storage, and waiting for the prediction results back.
Then, a cloud function written node.js is implemented onto the cloud, and the cloud function will be automatically triggered whenever an image is uploaded to the specific storage bucket. Then the machine learning Engine will generate predictions and saved the results in the cloud database.

Finally, a listener is created on the cloud database. Once new data has been added in the database, the client application will fetch the data which contains

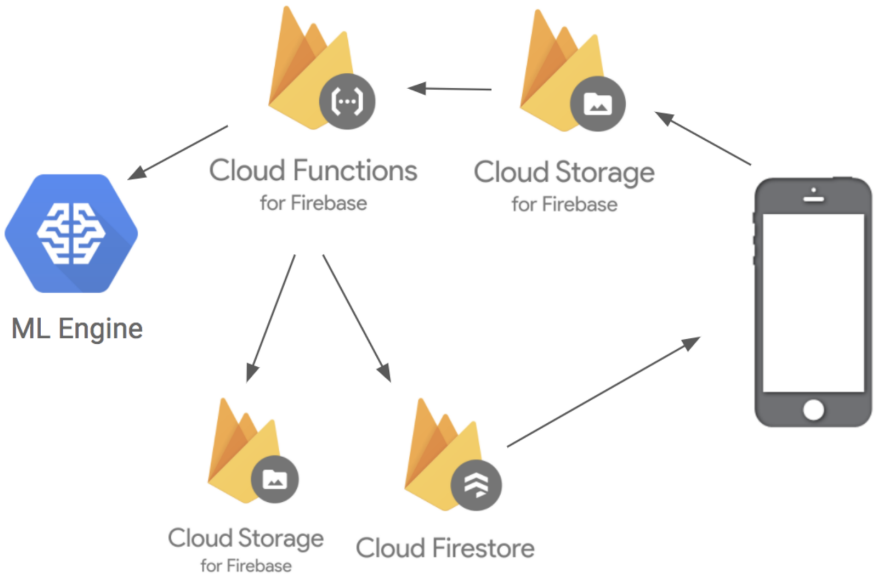


Figure 23: Google Cloud Communication

information including the new outlined image with the bounding boxes around, along with the predicted name and confidence from the Google cloud.

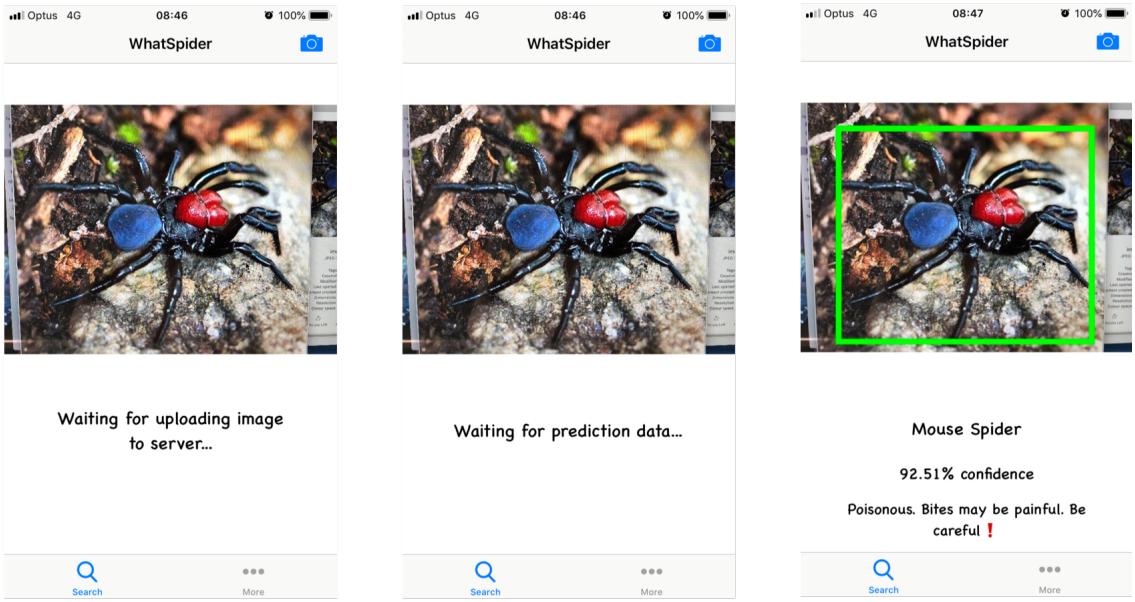
6.4 UI Design

A user-friendly interface is very crucial for the software project. And the process should be smooth and simple for users to operate. Taken these into considerations, the interface is designed as simple as possible but also meet the basic requirements in this project.

Since the purpose of the project is to detect whether the spider is poisonous or not and let the users know more about the spider. Therefore, WIKI API is used in this project for the additional information for users to explore and learn more about the spider information.

The client application after successfully receiving prediction information will show three recognition messages in the main interface:

(i) The prediction picture with bounding box



Step 1:User Upload the photo Step 2:Wait for the results Step 3: Prediction Results

Figure 24: User Process

(ii) The prediction spider name

(iii) The prediction confidence.

Figure 25: Introduction about the predicted spider

Once the kind of the spider is predicted, its corresponding poisonous level can be known from previous spider species investigation. The advice is given in Section 1.4.

7 Further Work

Due to the limited resources and time, there are many aspects the software could be improved.

As discussed in the introduction, the dataset quality and amount greatly affect the model performance. Therefore, despite various sources of data online, qualified and high resolution pictures of spiders are still needed to improve the accuracy of the model. In order to get a more valuable dataset, professional organizations such as the Australian Museum can improve the quality of the dataset.

On the other side, SSD Mobilenet V1 FPN has a significant performance in detecting large objects. However, spiders are small creatures. To some extent, it increases the difficulty of prediction for users using the mobile application to take a clear picture of an object. Therefore, other models which are good at recognizing small objects with higher accuracy can be taken into consideration for further development. For example, there is a Traffic light detection and Recognition [3] and Face detection for Surveillance [24] are two successful examples for small multiple object prediction.

8 Conclusion

In this project, the architecture of existing deep learning models and their effectiveness applied in object detection are investigated. An IOS application is implemented for detecting Australian spiders. To begin with, massive data of spider image were crawled from three major photo gallery online (Bing, Flickr, and Google). And then, these images are preprocessed to get relatively pure data set (17 spider species and 3 confusing species). Finally, the SSD Mobilenet V1

FPN model is retrained based on the spider dataset and applied the model to the IOS development.

In practice, some spiders which are very similar to each other are the main factor influencing the prediction accuracy. Notably, the SSD model is not good at detecting the small object, although it has a preeminent speed compared to existing models. Therefore, a good quality of training dataset, as well as a effective detection framework can greatly enhance the performance of the custom model in detecting objects even on complex scenarios.

9 References

[1] Y. Bengio, A. Courville and P. Vincent, 'Representation learning: A review and new perspectives', *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] L. Zhang, J. Tan, D. Han and H. Zhu, 'From machine learning to deep learning: Progress in machine intelligence for rational drug discovery', *Drug discovery today*, vol. 22, no. 11, pp. 1680–1685, 2017.

[4] A. Moujahid. (2016). A practical introduction to deep learning with caffe and python, [Online]. Available: http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/.

[5] Y. LeCun, Y. Bengio and G. Hinton, 'Deep learning', *Nature*, vol. 521, no. 7553, p. 436, 2015.

[6] Supervise.ly. (2017). Big challenge in deep learning: Training data, [Online]. Available: https://hackernoon.com/%EF%B8%8F-big-challenge-in-deep-learning-training-data-31a88b97b282.

[7] R. Pryss, M. Reichert, W. Schlee, M. Spiliopoulou, B. Langguth and T. Probst, 'Differences between android and ios users of the trackyourtinnitus mobile crowdsensing mhealth platform', in *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, 2018, pp. 411–416.

[8] P. Escoubas, S. Diochot and G. Corzo, 'Structure and pharmacology of spider venom neurotoxins', *Biochimie*, vol. 82, no. 9-10, pp. 893–907, 2000.

[9] M. H. Greenstone, 'Determinants of web spider species diversity: Vegetation structural diversity vs. prey availability', *Oecologia*, vol. 62, no. 3, pp. 299–304, 1984.

[10] W. S. Catalog, 'World spider catalog', *Natural History Museum Bern*, 2015.

[11] R. Foelix, *Biology of spiders*. OUP USA, 2011.

[12] S. Sutherland and J. Trinca, 'Survey of 2144 cases of red-back spider bites: Australia and new zealand, 1963-1976', *Medical Journal of Australia*, vol. 2, no. 14, pp. 620–623, 1978.

[13] S. C. Wong, A. Gatt, V. Stamatescu and M. D. McDonnell, 'Understanding data augmentation for classification: When to warp?', in *2016 international conference on digital image computing: Techniques and applications (DICTA)*, IEEE, 2016, pp. 1–6.

[14] S. J. Pan and Q. Yang, 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[15] V. Ilievski, C. Musat, A. Hossmann and M. Baeriswyl, 'Goal-oriented chatbot dialog management bootstrapping with transfer learning', *ArXiv preprint arXiv:1802.00500*, 2018.

[16] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, 'Speed/accuracy trade-offs for modern convolutional object detectors', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[17] D. Erhan, C. Szegedy, A. Toshev and D. Anguelov, 'Scalable object detection using deep neural networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2147–2154.

[18] C. Szegedy, S. Reed, D. Erhan, D. Anguelov and S. Ioffe, 'Scalable, high-quality object detection', *ArXiv preprint arXiv:1412.1441*, 2014.

[19] S. Ren, K. He, R. Girshick and J. Sun, 'Faster r-cnn: Towards real-time object detection with region proposal networks', in *Advances in neural information processing systems*, 2015, pp. 91–99.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, 'Ssd: Single shot multibox detector', in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, 'Mobilenets: Efficient convolutional neural networks for mobile vision applications', *ArXiv preprint arXiv:1704.04861*, 2017.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, 'Feature pyramid networks for object detection', in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[23] N. Saravanan, A. Mahendiran, N. V. Subramanian and N. Sairam, 'An implementation of rsa algorithm in google cloud using cloud sql', *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 19, pp. 3574–3579, 2012.

[24] G. L. Foresti, C. Micheloni, L. Snidaro and C. Marchiol, 'Face detection for visual surveillance', in *12th International Conference on Image Analysis and Processing, 2003. Proceedings.*, IEEE, 2003, pp. 115–120.

10 Appendices

Source Code: https://github.com/SiteHuang/COMP90055-Spider-Recognition
Demo Video Link: https://youtu.be/NJM5_-XzV90