ESKİŞEHİR TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

***Bhealth*: AI Supported Honey Bee Healthy Analysis**

Sefa Salim BOZDAĞ
Rüştü Efe UZUN

A Bachelor of Science Project

Department of Computer Engineering

June 2023

Bhealth: AI Supported Honey Bee Healthy Analysis

by

Sefa Salim BOZDAĞ
(sefabozdag41@gmail.com)
Rüştü Efe UZUN
(reuzun@outlook.com)

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Engineering is approved by the following scientific committee members.

Date of Approval	**:**
Member (Advisor)	**: Assist. Prof. Dr. Burcu YILMAZEL**
Member	**: Assist. Prof. Dr. Sema CANDEMİR**
Member	**: Lect. Emre KAÇMAZ**

ABSTRACT

Beekeeping has been a crucial issue since ancient times, as honey bees not only produce honey but also create more value than any other living thing by dispersing pollen among plants. In addition to that, these plants are mainly used by the medical industry and as industrial materials in factories. So that having healthy bees in our world is a very important matter for our world to be sustainable and livable for the next generations. Our study predominantly investigates the detection of the health problems of honey bees using their images taken through a user-friendly mobile application by machine learning methods.

In the study, we have compared the performance of two different methods for classifying the health of the bees, Logistic Regression and Convolutional Neural Network (CNN), on a well-known BeeImage Dataset that contains annotated honey bee images. Our experimental studies indicated that the CNN model outperformed the other method, so we have decided to use it in our system. Beyond this, we have augmented the image dataset to check the effects of data augmentation techniques on the classifier performances. However, it does not affect performance. We have also created a server using Django which uses our classifier model under the hood, and in the mobile application development, we have used Flutter to support multiple different mobile operating systems from a single codebase. Our study has the potential to lead other future research. In the future, our application can be empowered with computer vision, can be used with Internet of Things (IoT) devices for bee hives, and can be used to create massive data in finding out which locations would be more suitable for beekeeping.

Keywords: Beekeeping, Sustainability, Climate Change, Image Processing, Deep Learning.

ÖZET

Arıcılık, antik çağlardan bu yana önemli bir konu olmuştur çünkü bal arıları sadece bal üretmekle kalmaz, bitkiler arasında polen yayarak diğer canlılardan daha fazla değer yaratırlar. Ayrıca, bu bitkiler çoğunlukla tıp endüstrisinde ve fabrikalarda endüstriyel malzeme olarak kullanılmaktadır. Dolayısıyla, sağlıklı arılara sahip olmak, dünyamızın gelecek nesiller için sürdürülebilir ve yaşanabilir olması açısından son derece önemli bir konudur. Çalışmamız, kullanıcı dostu bir mobil uygulama aracılığıyla elde edilen arı görüntüleri kullanılarak makine öğrenimi yöntemleriyle arıların sağlık sorunlarının tespitini incelemektedir.

Çalışmamızda, iyi bilinen BeeImage Veri Kümesi'nde yer alan etiketli arı görüntülerini kullanarak arıların sağlık durumunu sınıflandırmak için iki farklı yöntem olan Lojistik Regresyon ve Evrişimsel Sinir Ağı (CNN) yöntemlerinin performansını karşılaştırdık. Deneysel çalışmalarımız, CNN modelinin diğer yöntemden daha başarılı olduğunu gösterdi, bu yüzden sistemimizde bunu kullanmaya karar verdik. Bunun ötesinde, sınıflandırıcı performansları üzerinde veri artırma tekniklerinin etkisini kontrol etmek için görüntü veri kümesini artırdık. Ancak, veri artırma performansı etkilemedi. Ayrıca, sınıflandırıcı modelimizi kullanan bir Django sunucusu oluşturduk ve mobil uygulama geliştirme sürecinde Flutter'ı kullanarak tek bir kod tabanından birden çok farklı mobil işletim sistemini destekledik. Çalışmamız, diğer gelecekteki araştırmalara öncülük edebilecek potansiyele sahiptir. Gelecekte, uygulamamız bilgisayar görüşü ile güçlendirilebilir, arı kovanları için Nesnelerin İnterneti (IoT) cihazlarıyla kullanılabilir ve arıcılık için daha uygun bölgelerin belirlenmesinde büyük veri oluşturmak için kullanılabilir.

Anahtar Kelimeler: Arıcılık, Sürdürülebilirlik, İklim Değişikliği, Görüntü İşleme, Derin Öğrenme

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our advisor Assist. Prof. Dr. Burcu YILMAZEL for their invaluable support and guidance throughout the completion of this thesis. We are especially grateful for their constructive suggestions and topic recommendations on bees and climate change, which enabled us to bring this project to life. Their profound knowledge, insightful directions, and inspiring attitude have greatly enhanced the quality and success of this work. We are truly thankful for their mentorship.

Furthermore, we would like to extend our thanks to TÜBİTAK (The Scientific and Technological Research Council of Turkey) for their support in our project, which they deemed worthy of assistance under the TÜBİTAK 2209 A program. Their financial and moral support has provided us with the opportunity to conduct this research. The efforts and contributions of TÜBİTAK in supporting scientific projects have greatly contributed to the success of this study. We are grateful to TÜBİTAK for their support.

CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
ML	Machine Learning
LR	Logistic Regression

1. INTRODUCTION

1.1 Project Definition

Bhealth is an application that identifies the health situation of the honeybees. To be able to process the identification dataset is required. We decided to use The BeeImage Dataset: Annotated Honey Bee Images [1]. Different kinds of algorithms have been tested to be able to have the most successful machine learning model. Moreover, data augmentation techniques have been applied to boost the success ratio of ML model. To be able to eveluate the ML model also a mobile application is created.

1.2 Project Goal

The main purpose of the proposed project is to determine whether a given image belongs to honey bees, by using image processing and machine learning techniques, and if it belongs to honey bees, by determining the health status of the honey bee in the image and the type of disease it has been exposed to; It is to present the transaction results to the user through the mobile application with a clear, simple and useful interface and to ensure that each newly added image improves the analysis results.

In this context;

1) Examining the performance of different algorithms to be developed with classical machine learning methods in determining the health status of the bee and the bee over bee images,
2) Examining the performance of different algorithms to be developed with deep learning methods in determining the health status of the bee and the bee over bee images,
3) Comparing the performance results obtained with classical machine learning methods with the results developed by deep learning methods,
4) Making the most successful algorithm available to beekeepers with a user-friendly mobile application.

are the main objectives of the project.

2. TECHNOLOGIES

2.1 Python [2]

Python is a versatile and widely-used programming language known for its simplicity and readability. It has a large and active community, making it ideal for various applications. Python is extensively used in data analysis, artificial intelligence, web development, scientific computing, and more.

2.2 TensorFlow [3]

TensorFlow is a powerful open-source machine learning framework developed by Google. It provides a comprehensive ecosystem for building and deploying machine learning models. TensorFlow supports both deep learning and traditional machine learning techniques, making it a popular choice for a wide range of applications.

2.3 Scikit-learn [4]

Scikit-learn is a free and open-source machine learning library for Python. It offers a diverse range of algorithms for classification, regression, clustering, and dimensionality reduction tasks. With its easy-to-use API and extensive documentation, Scikit-learn is a popular choice for both beginners and experienced machine learning practitioners.

2.4 Django [5]

Django is a high-level Python web framework that enables rapid development and clean design. It follows the model-view-controller (MVC) architectural pattern and promotes the principle of Don't Repeat Yourself (DRY). Django provides a robust set of tools and features, including an ORM (Object-Relational Mapping) for database interactions, authentication system, URL routing, template engine, and more.

2.5 Flutter [6]

Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop platforms. Using the Dart programming language, Flutter allows developers to create visually appealing and performant user interfaces. It offers a rich set of pre-built widgets, hot-reload functionality, and a reactive framework, making app development efficient and enjoyable.

3. METHODOLOGY

Based on the recommendation of our advisor, we conducted a literature review on bees and climate change for our project. We discovered the Annotated Bee Dataset on Kaggle, which we determined to be suitable for our project concept. As a result, we decided to develop our project using this dataset. Shortly after, we prepared the process flowchart of our project on figure 3.1.

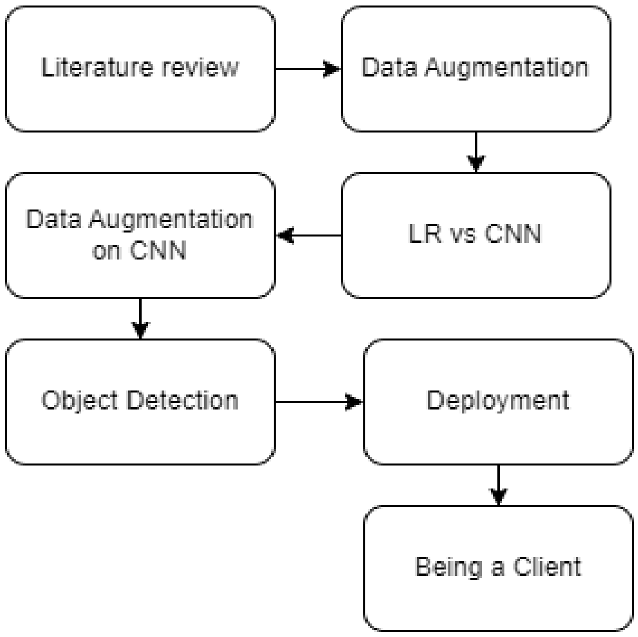


Figure 3.1: Flowchart of the project

3.1 Literature Review

While conducting a literature review on bees and climate change, as mentioned earlier, we came across a dataset called Annotated Bee Dataset on Kaggle. This dataset contains labeled images belonging to 5 different health classes. The classes include ant problem, few varroa hive beetles, healthy, varroa small hive beetles, and hive being robbed. Example images for each class are shown in Figure 3.2.

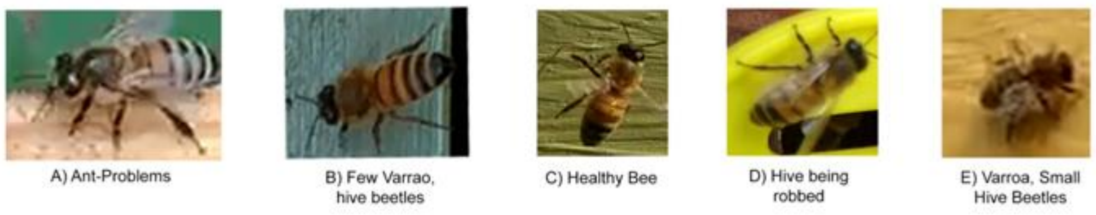


Figure 3.2: Example images for each class

3.2 Data Augmentation

The original data we have is as shown in Figure 3.2. In order to enhance the training performance of our classification model, balance the dataset, and facilitate the classification of bees, we have augmented the dataset using methods such as rotation and cropping. To determine the effectiveness of data augmentation, we will train and test the selected best classification model using both the original and augmented data, and proceed with the dataset that performs better accordingly. The main objective here is to better accommodate the quality of images uploaded by users and provide accurate results for images of various qualities and angles. Example images generated through data augmentation are shown in Figure 3.3.

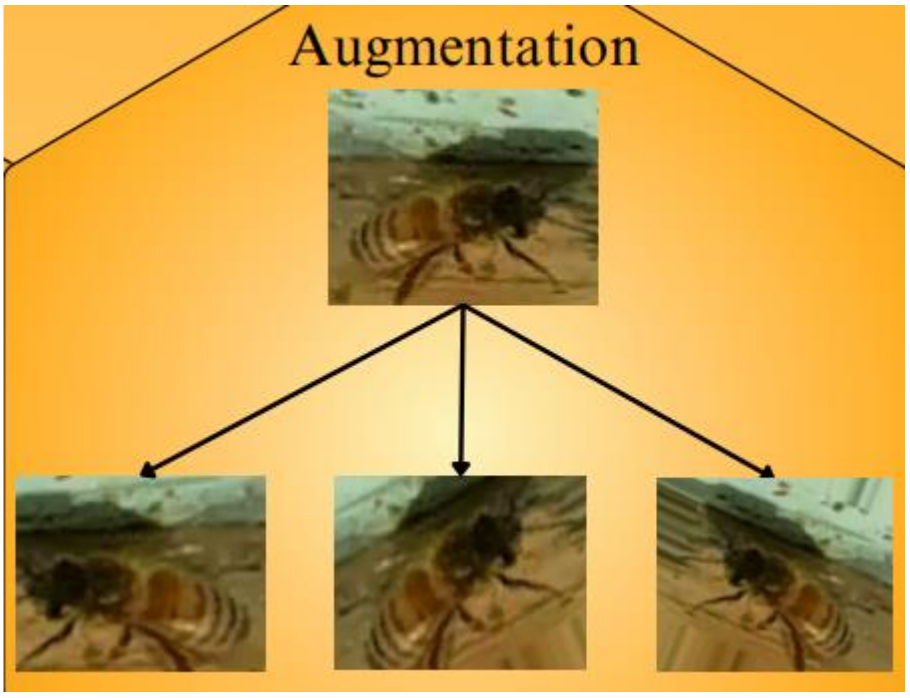
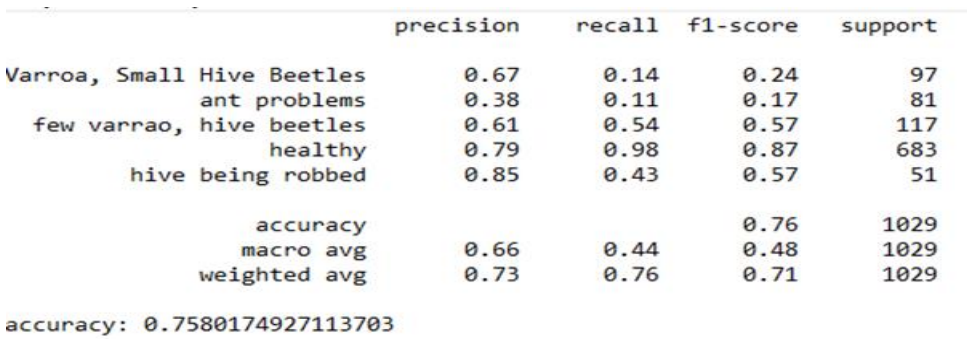


Figure 3.3: Example images generated through data augmentation

3.3 LR vs CNN

After data augmentation, we ended up with two datasets: the augmented dataset and the original dataset. We used two classification methods to train and compare these datasets. The first method is Logistic Regression (LR), which is a traditional machine learning method, and the second one is Convolutional Neural Networks (CNN), which is a modern deep learning method. Firstly, we trained a model with each method using the original dataset. The classification results for LR trained with the original data are shown in Figure 3.4, and the confusion matrix is provided in Figure 3.5. Similarly, the classification results for CNN trained with the original data are shown in Figure 3.6, and the confusion matrix is given in Figure 3.7.



```
                            precision    recall  f1-score   support

Varroa, Small Hive Beetles       0.67      0.14      0.24        97
              ant problems       0.38      0.11      0.17        81
   few varrao, hive beetles      0.61      0.54      0.57       117
                   healthy       0.79      0.98      0.87       683
         hive being robbed       0.85      0.43      0.57        51

                  accuracy                           0.76      1029
                 macro avg       0.66      0.44      0.48      1029
              weighted avg       0.73      0.76      0.71      1029

accuracy: 0.7580174927113703
```

Figure 3.4: Classification report of LR with original data

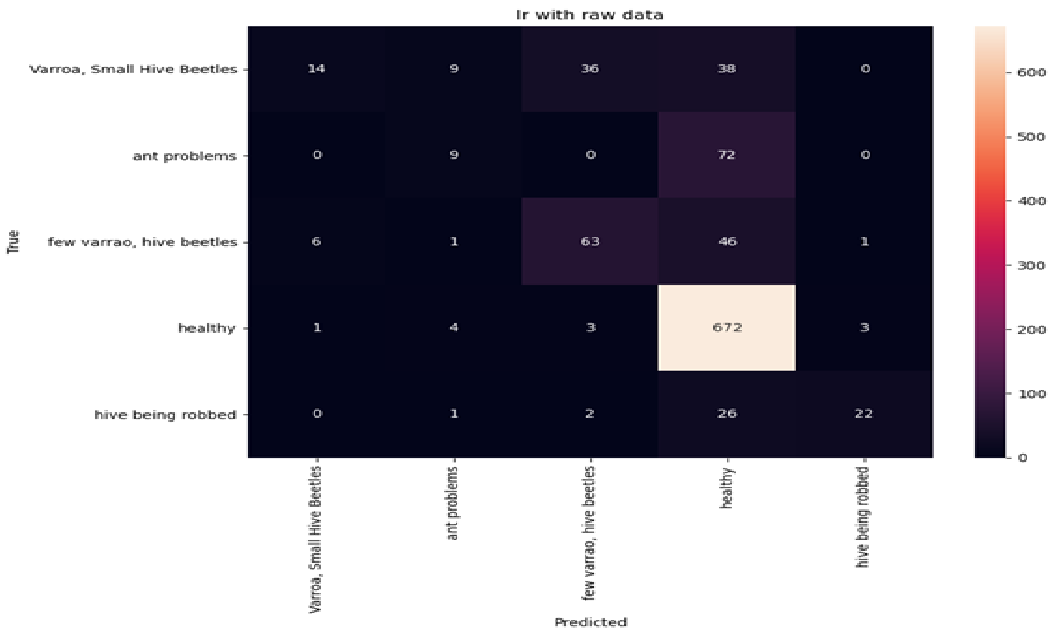
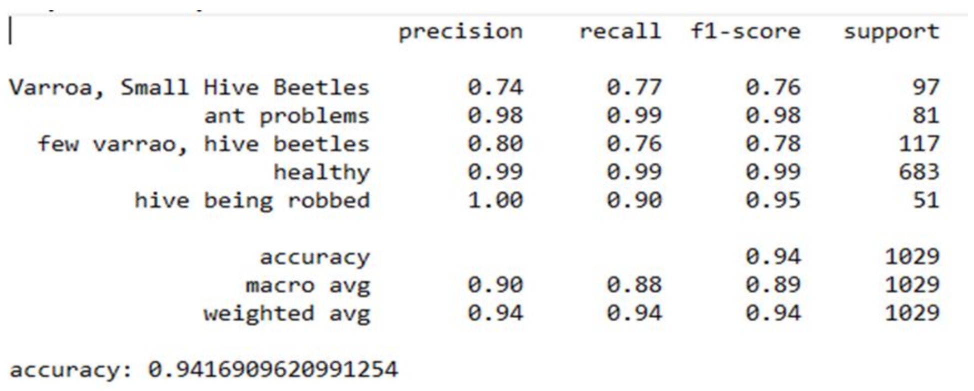


Figure 3.5: Confusion matrix of LR with original data



```
                         precision    recall  f1-score   support

Varroa, Small Hive Beetles       0.74      0.77      0.76        97
              ant problems       0.98      0.99      0.98        81
    few varrao, hive beetles     0.80      0.76      0.78       117
                   healthy       0.99      0.99      0.99       683
         hive being robbed       1.00      0.90      0.95        51

                  accuracy                           0.94      1029
                 macro avg       0.90      0.88      0.89      1029
              weighted avg       0.94      0.94      0.94      1029

accuracy: 0.9416909620991254
```

Figure 3.6: Classification report of CNN with original data

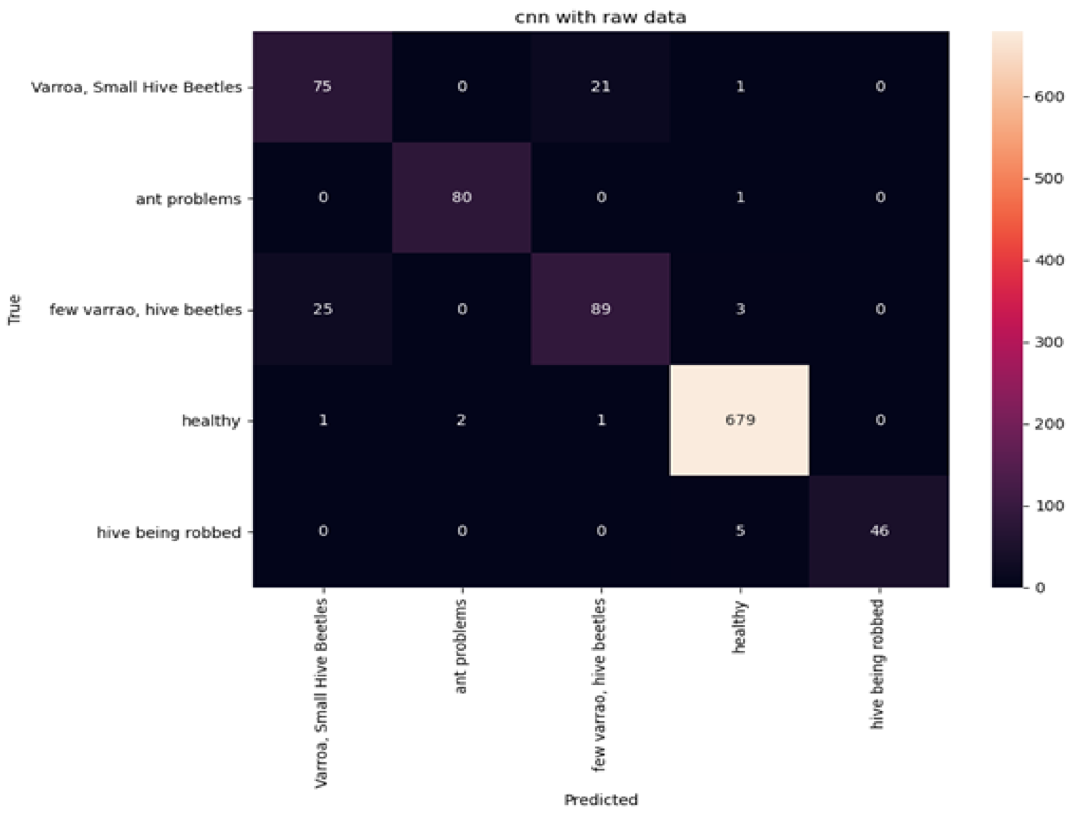
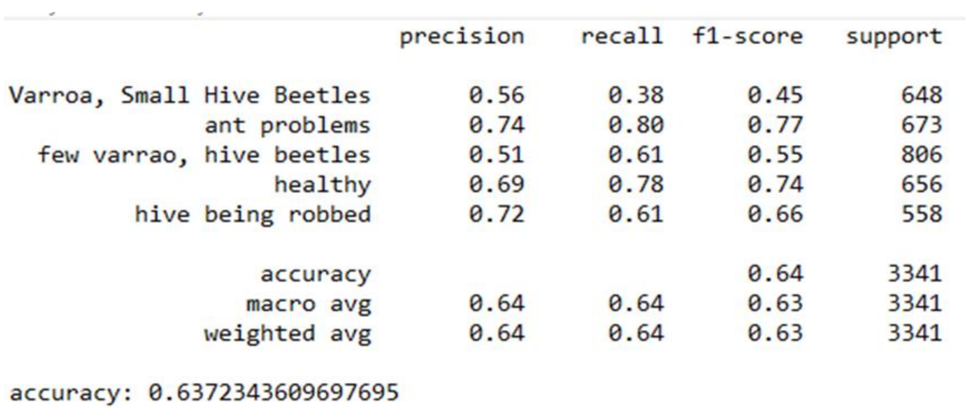


Figure 3.7: Confusion matrix of CNN with original data

By examining the classification reports, it can be observed that the CNN model outperforms the LR model, which is further supported by the patterns in the confusion matrices. The patterns in the confusion matrix for the CNN model are more organized compared to the confusion matrix for the LR model. From this, it is evident that the

CNN model performs better for the original data. The same comparison was made using the augmented data as well. The classification report for the LR model trained with augmented data is shown in Figure 3.8, and the confusion matrix is provided in Figure 3.9. Similarly, the classification report for the CNN model trained with augmented data is shown in Figure 3.10, and the confusion matrix is given in Figure 3.11.



```
                           precision    recall  f1-score   support

Varroa, Small Hive Beetles      0.56      0.38      0.45       648
              ant problems      0.74      0.80      0.77       673
    few varrao, hive beetles    0.51      0.61      0.55       806
                   healthy      0.69      0.78      0.74       656
          hive being robbed     0.72      0.61      0.66       558

                  accuracy                          0.64      3341
                 macro avg      0.64      0.64      0.63      3341
              weighted avg      0.64      0.64      0.63      3341

accuracy: 0.6372343609697695
```

Figure 3.8: Classification report of LR with augmented data

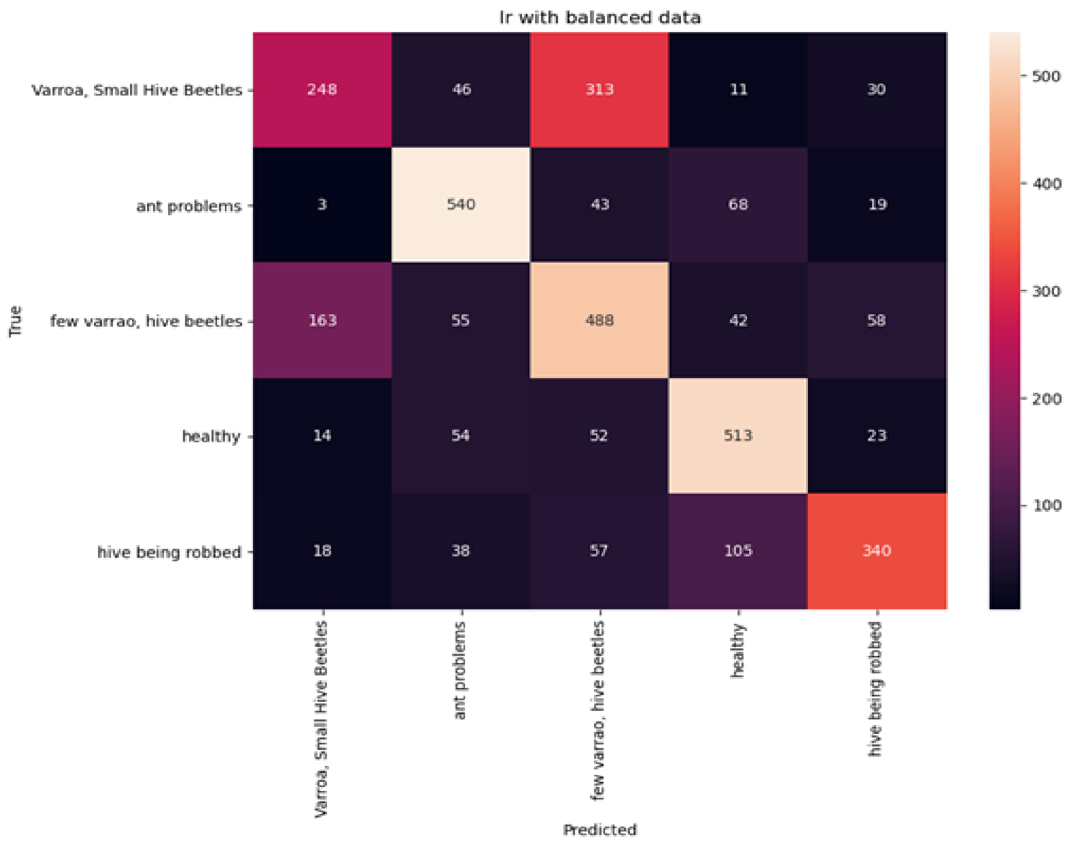
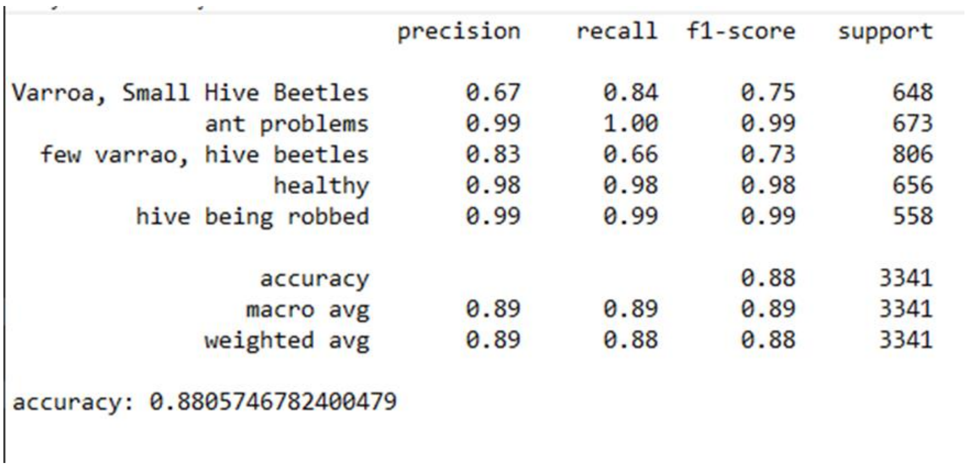


Figure 3.9: Confusion matrix of LR with augmented data



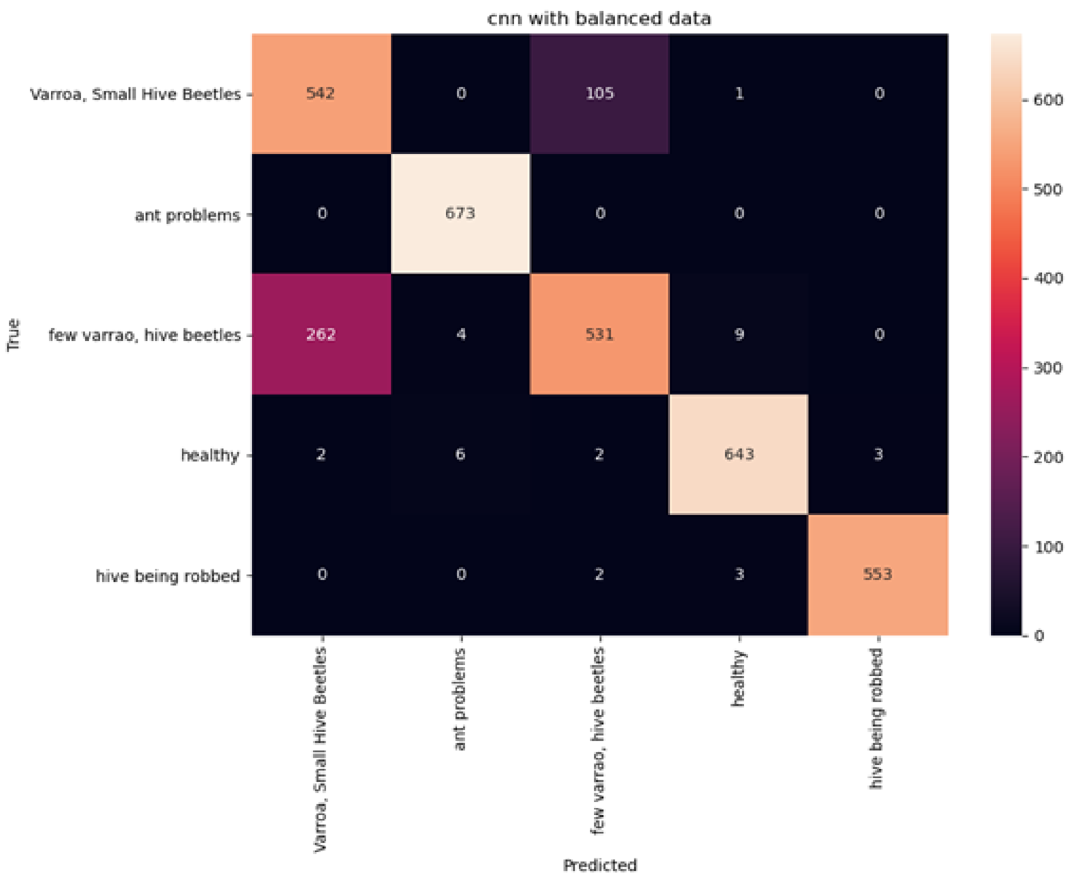
```
                              precision    recall  f1-score   support

Varroa, Small Hive Beetles         0.67      0.84      0.75       648
              ant problems         0.99      1.00      0.99       673
  few varrao, hive beetles         0.83      0.66      0.73       806
                   healthy         0.98      0.98      0.98       656
         hive being robbed         0.99      0.99      0.99       558

                  accuracy                             0.88      3341
                 macro avg         0.89      0.89      0.89      3341
              weighted avg         0.89      0.88      0.88      3341

accuracy: 0.8805746782400479
```

Figure 3.10: Classification report of CNN with augmented data



True \ Predicted	Varroa, Small Hive Beetles	ant problems	few varrao, hive beetles	healthy	hive being robbed
Varroa, Small Hive Beetles	542	0	105	1	0
ant problems	0	673	0	0	0
few varrao, hive beetles	262	4	531	9	0
healthy	2	6	2	643	3
hive being robbed	0	0	2	3	553

Figure 3.11: Confusion matrix of CNN with augmented data

According to the classification results, the CNN model has a higher accuracy rate compared to the LR model, and when examining the confusion matrices, it can be observed that the pattern is more organized on the CNN side. This indicates that the CNN, which is a modern deep learning method, outperforms the LR, which is a traditional machine learning method, both in the experiments conducted with the original and augmented data. A summary of this section, represented by a bar graph, is provided in Figure 3.12.

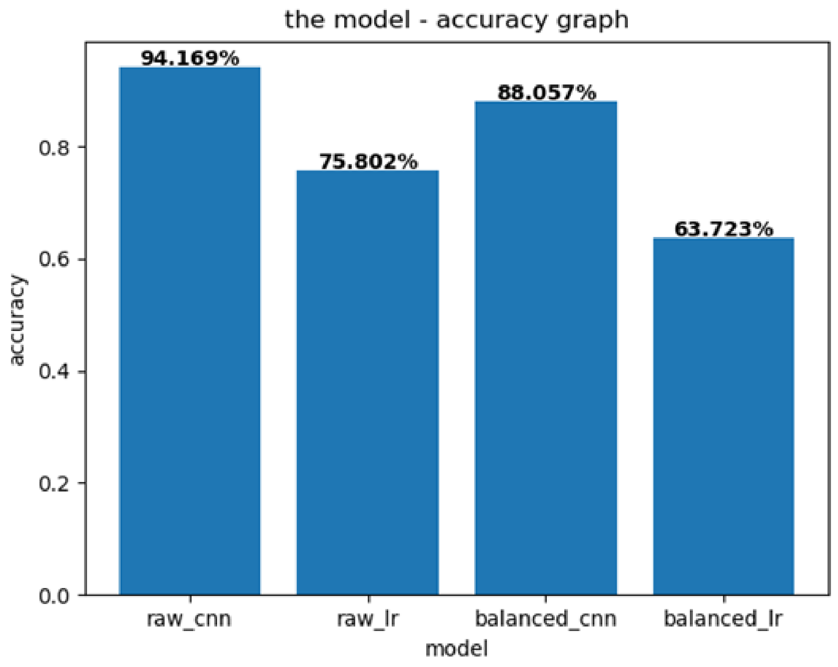


Figure 3.12: Bar graph of accuracy of all combinations in LR vs CNN

3.4 Data Augmentation on CNN

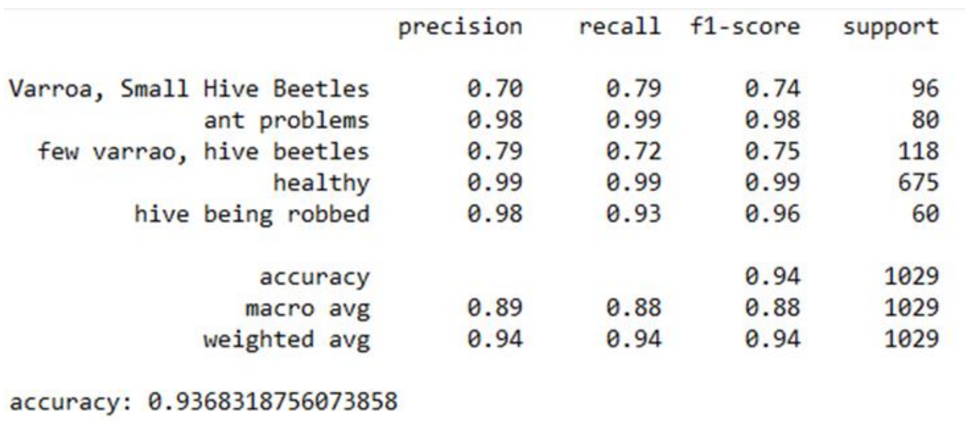
Data augmentation is a technique commonly used in machine learning and deep learning to artificially increase the size of a training dataset by creating modified or synthetic versions of existing data samples. It offers several advantages in improving model performance. Firstly, data augmentation helps to mitigate overfitting by introducing diversity and reducing the risk of the model memorizing the training examples. It allows the model to generalize better to unseen data. Secondly, data augmentation increases the robustness of the model by exposing it to a wider range of variations, such as rotations, translations, flips, and distortions. This enables the model to learn invariant features and become more resilient to noise and variations in the input data. Additionally, data augmentation can address class imbalance issues by creating additional samples for underrepresented classes, which leads to more balanced training.

However, data augmentation also has some limitations and potential drawbacks. One major disadvantage is the possibility of introducing unrealistic or incorrect samples during the augmentation process, which can lead to misleading patterns and negatively impact model performance. It requires careful selection of augmentation techniques and parameters to ensure that the generated samples retain the desired characteristics of the original data. Moreover, data augmentation can significantly increase the computational and storage requirements, as the augmented dataset can be much larger than the original dataset. This can impact training time and resource utilization. Lastly, data augmentation techniques may not be equally effective for all types of datasets or tasks. The suitability of specific augmentation methods depends on the nature of the data and the objectives of the task, and experimentation is often needed to determine the most effective augmentation strategy for a given problem.

Considering these factors, the impact of data augmentation on the CNN model was further examined in this section. Firstly, we have two datasets available: the original dataset and the augmented dataset. In this section, we tested four different scenarios, which are as follows:

1) CNN model trained with the original data and tested with the original data,
2) CNN model trained with the augmented data and tested with the original data,
3) CNN model trained with the original data and the augmented data,
4) CNN model trained with the augmented data and tested with the augmented data.

Firstly, the two scenarios where the original data was tested were compared. The classification report for the CNN model trained with the original data and tested with the original data is shown in Figure 3.13, and the confusion matrix is shown in Figure 3.14. Additionally, the classification report for the CNN model trained with the augmented data and tested with the original data is presented in Figure 3.15, while the corresponding confusion matrix is displayed in Figure 3.16.



```
                          precision    recall  f1-score   support

Varroa, Small Hive Beetles       0.70      0.79      0.74        96
              ant problems       0.98      0.99      0.98        80
   few varrao, hive beetles       0.79      0.72      0.75       118
                   healthy       0.99      0.99      0.99       675
         hive being robbed       0.98      0.93      0.96        60

                  accuracy                           0.94      1029
                 macro avg       0.89      0.88      0.88      1029
              weighted avg       0.94      0.94      0.94      1029

accuracy: 0.9368318756073858
```

Figure 3.13: Classification report of CNN trained with original, tested with original data

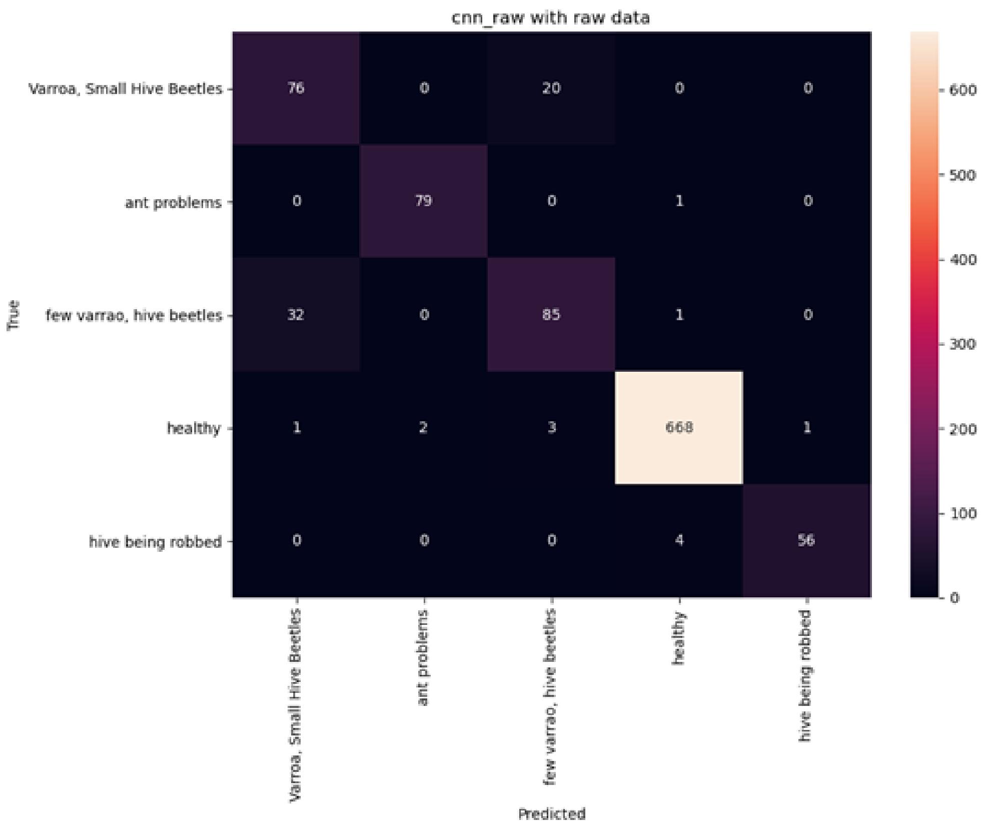
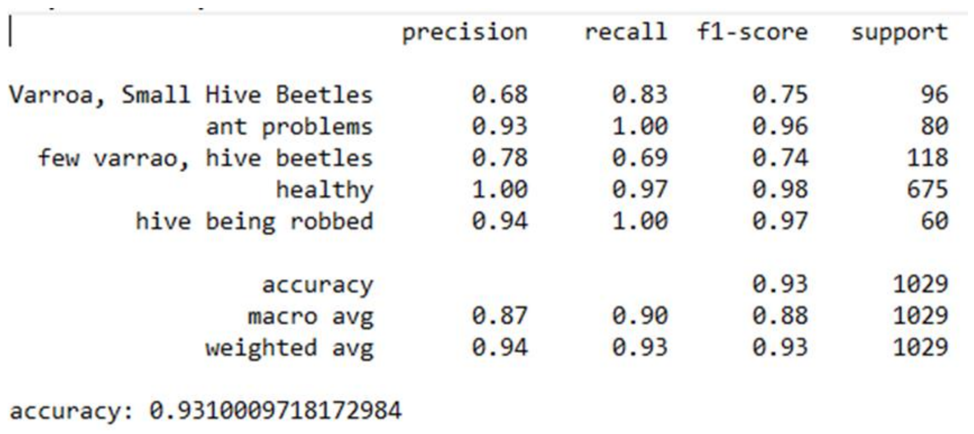


Figure 3.14: Confusion matrix of CNN trained with original, tested with original data



```
                              precision    recall  f1-score   support

Varroa, Small Hive Beetles       0.68      0.83      0.75        96
              ant problems       0.93      1.00      0.96        80
  few varrao, hive beetles       0.78      0.69      0.74       118
                   healthy       1.00      0.97      0.98       675
         hive being robbed       0.94      1.00      0.97        60

                  accuracy                           0.93      1029
                 macro avg       0.87      0.90      0.88      1029
              weighted avg       0.94      0.93      0.93      1029

accuracy: 0.9310009718172984
```

Figure 3.15: Classification report of CNN trained with augmented, tested with original data

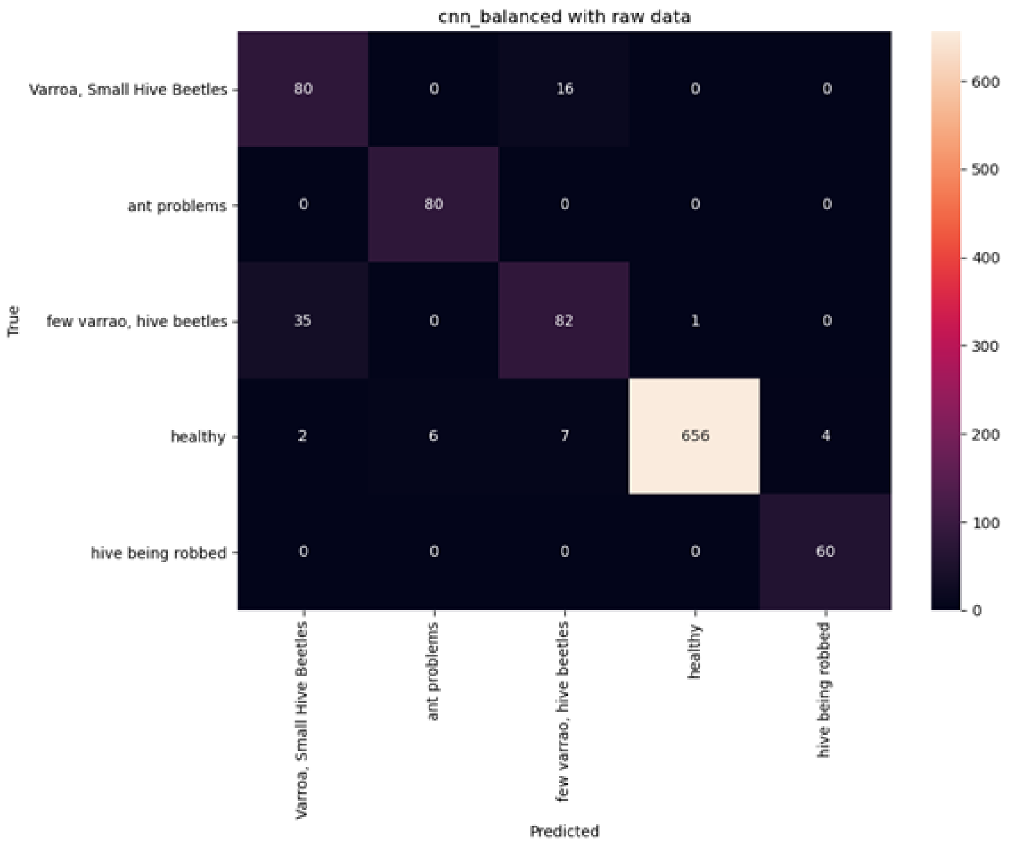
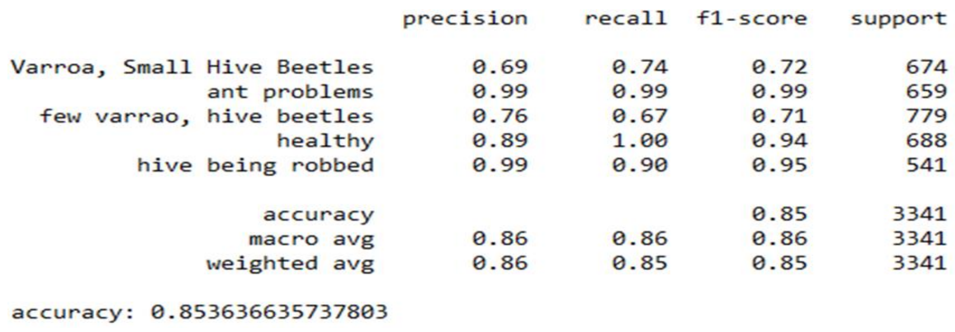


Figure 3.16: Confusion matrix of CNN trained with augmented, tested with original data

As observed, when tested with the original data, it can be concluded that the training dataset does not have an extreme impact on the outcome, and training the model with either the original or augmented dataset does not significantly affect the classification performance. To better understand the effect of data augmentation on the CNN model, the next stage of the experiment compared models trained with the original data and tested with the augmented data, as well as models trained and tested with the augmented data. The classification report for the CNN model trained with the original data and tested with the augmented data is presented in Figure 3.17, and the corresponding confusion matrix is shown in Figure 3.18. Furthermore, the classification report for the CNN model trained and tested with the augmented data is displayed in Figure 3.19, while the confusion matrix can be seen in Figure 3.20.



```
                           precision    recall  f1-score   support

Varroa, Small Hive Beetles      0.69      0.74      0.72       674
              ant problems      0.99      0.99      0.99       659
  few varrao, hive beetles      0.76      0.67      0.71       779
                   healthy      0.89      1.00      0.94       688
         hive being robbed      0.99      0.90      0.95       541

                  accuracy                          0.85      3341
                 macro avg      0.86      0.86      0.86      3341
              weighted avg      0.86      0.85      0.85      3341

accuracy: 0.853636635737803
```

Figure 3.17: Classification report of CNN trained with original, tested with augmented data

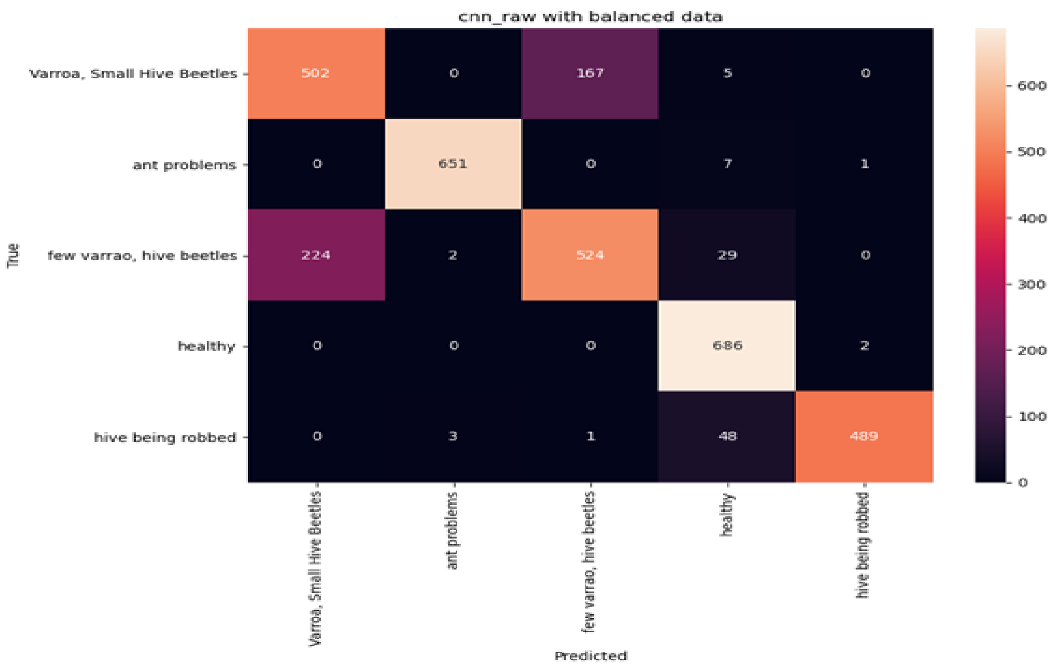
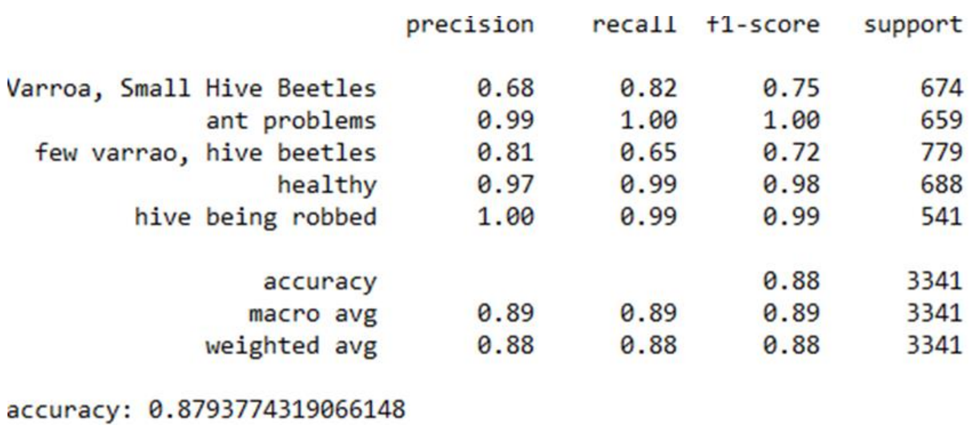


Figure 3.18: Confusion matrix of CNN trained with original, tested with augmented data



```
                          precision    recall  f1-score   support

Varroa, Small Hive Beetles       0.68      0.82      0.75       674
              ant problems       0.99      1.00      1.00       659
   few varrao, hive beetles      0.81      0.65      0.72       779
                    healthy      0.97      0.99      0.98       688
          hive being robbed      1.00      0.99      0.99       541

                   accuracy                          0.88      3341
                  macro avg      0.89      0.89      0.89      3341
               weighted avg      0.88      0.88      0.88      3341

accuracy: 0.8793774319066148
```

Figure 3.19: Classification report of CNN trained with augmented, tested with augmented data

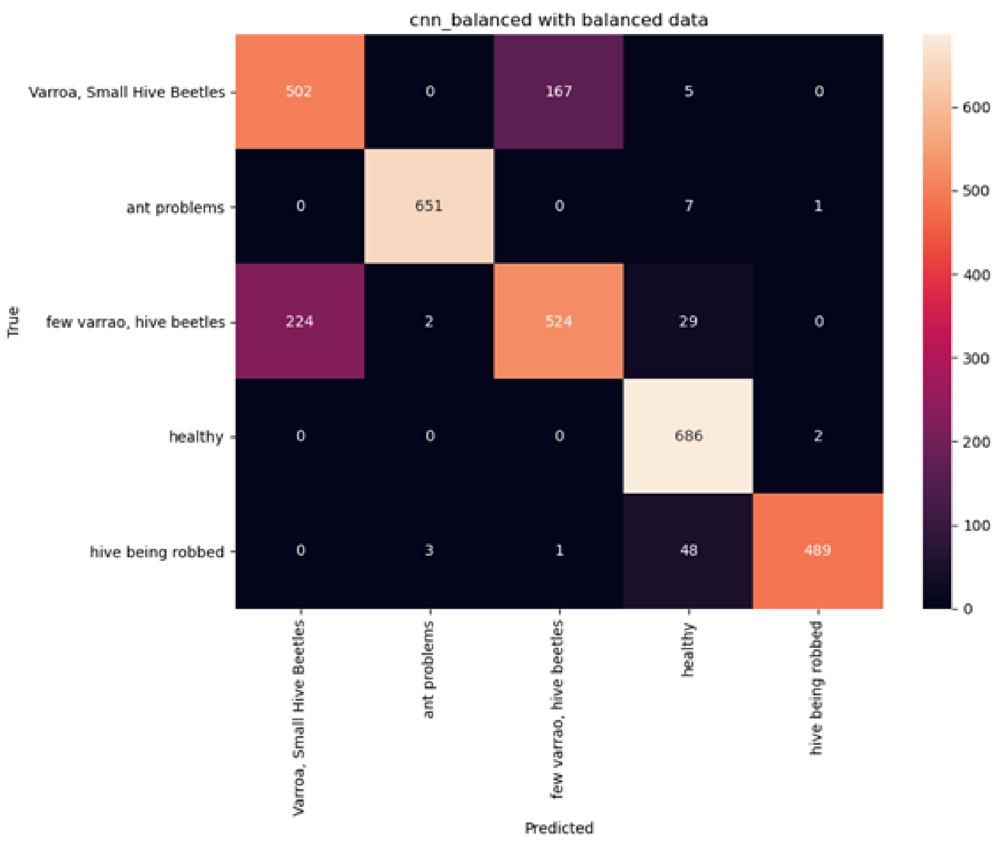


Figure 3.20: Confusion matrix of CNN trained with augmented, tested with augmented data

As evident from the results, when tested with the augmented data, the model's performance in classification is not significantly affected by whether it was trained on the original or augmented data. Considering all the scenarios, augmentation techniques such as image rotation and cropping have not demonstrated a noticeable positive or negative impact on the classification performance. A bar graph in Figure 3.21 presents the accuracies observed in all the cases during this experiment.

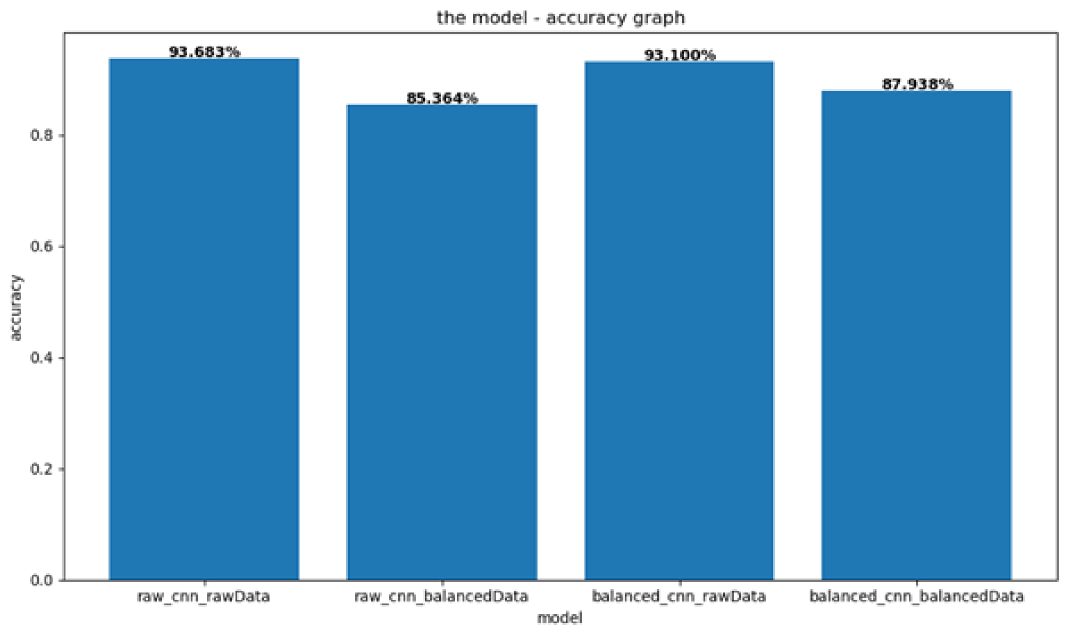


Figure 3.21: Accuracies of all cases

Taking these observations into account, it was found that when tested with augmented data, the models' performance was not significantly affected by whether the training data was the original or augmented dataset. However, it was observed from the confusion matrices of all models that certain classes within the Varroa category were frequently misclassified. Approximately 30% of the 'few varroa, hive beetles' class and 20-25% of the 'Varroa, small hive beetles' class were predicted as another Varroa class. On the other hand, the models performed well in the other classes. This pattern was also observed in the case of the original, unaugmented data. Although the discrepancy is not immediately noticeable when testing with unaugmented data, it still exists. For example, if there are around 5000 images in the unaugmented dataset and approximately 300 of them belong to the 'Varroa, small hive beetles' class, a misclassification rate of 20%

would result in 60 mispredictions. In the context of 5000 images, 60 mispredictions translate to a 1.2% accuracy loss. Similarly, in the 'few varroa, small hive beetles' class with 300 images, a 30% misclassification rate would lead to approximately 90 mispredictions, causing an accuracy loss of 1.8% against 5000 images.

The data augmentation used in this study aimed to balance the number of instances across all classes. The class with the highest number of instances is 'healthy' with around 3000 images. We augmented all classes to have approximately 3000 instances each, resulting in a total of 15000 instances, while the 'Varroa, small hive beetles' class specifically consists of 3000 instances. As in previous cases, if 20% of this class is misclassified as another Varroa class, 600 instances would be incorrectly predicted, leading to a 4% accuracy loss. Applying the same analysis to the 'few varroa, hive beetles' class, 900 out of 3000 instances would be misclassified, resulting in a 6% accuracy loss. While the misclassification of Varroa classes does not significantly affect the overall classification performance, it becomes more apparent when the ratio of Varroa instances to the total number of instances increases.

In the Kaggle study we referred to when constructing our model, a different data augmentation strategy was employed. They balanced the instances across all classes by using 600 instances per class, totaling 3000 instances. The accuracy rates obtained in that study ranged from 84% to 88%, which aligns with the results we achieved through data augmentation. The data augmentation strategy used in the Kaggle study also balanced the number of instances across all classes, resulting in a similar ratio of Varroa instances to the total number of instances, which explains the close accuracy rates obtained.

In conclusion, while the presence of data augmentation did not yield any noticeable positive or negative effects during training, the misclassification of Varroa classes was found to be a separate issue independent of data augmentation. Considering this, we have decided to proceed with the CNN model trained on augmented data.

3.5 Hyperparameter Configuration

In an attempt to improve the classification performance of the TensorFlow CNN model, various configurations were explored for two hyperparameters: epoch and batch size.

Epoch refers to the number of times the model iterates over the entire training dataset during training. Each epoch consists of one forward pass and one backward pass of all training samples.

Batch size determines the number of training samples processed in one forward and backward pass of the model. It affects the speed and memory requirements during training.

For the batch size, the following values were used: 64, 32, 16, 8, 4, 2, and 1. Training was conducted for each batch size using epoch values of 1, 10, 20, 30, 50, 100, and 250, respectively. The accuracy rate graphs for all batch sizes used for 1 epoch are shown in Figure 3.22, for 10 epochs in Figure 3.23, for 20 epochs in Figure 3.24, for 30 epochs in Figure 3.25, for 50 epochs in Figure 3.26, for 100 epochs in Figure 3.27, and for 250 epochs in Figure 3.28.

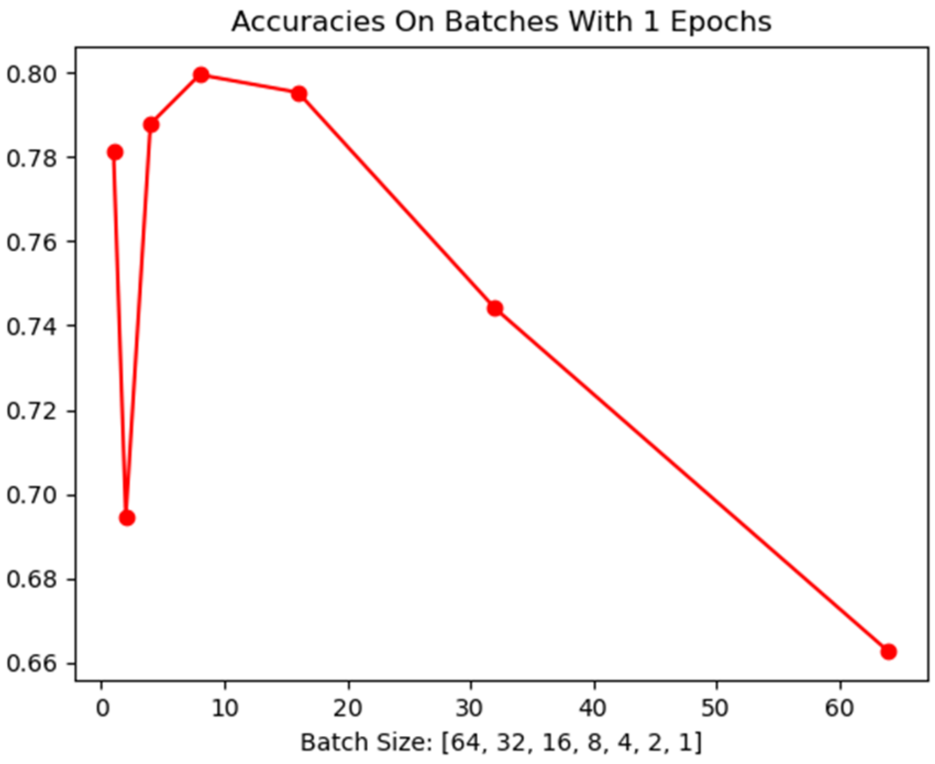


Figure 3.22: Accuracies on all batch-sizes with 1 epoch

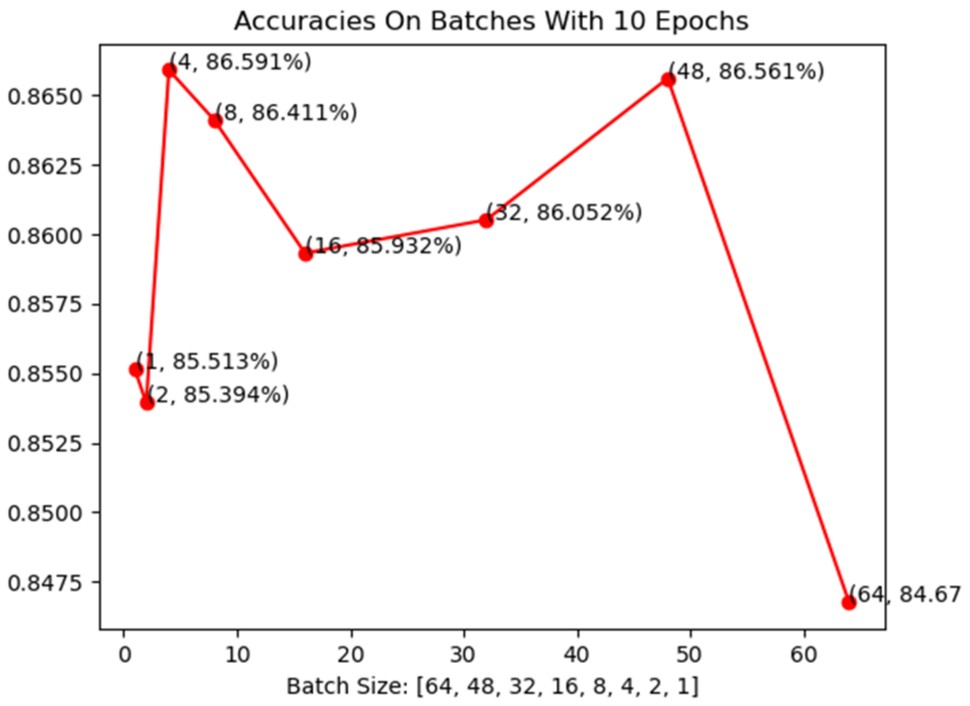


Figure 3.23: Accuracies on all batch-sizes with 10 epochs

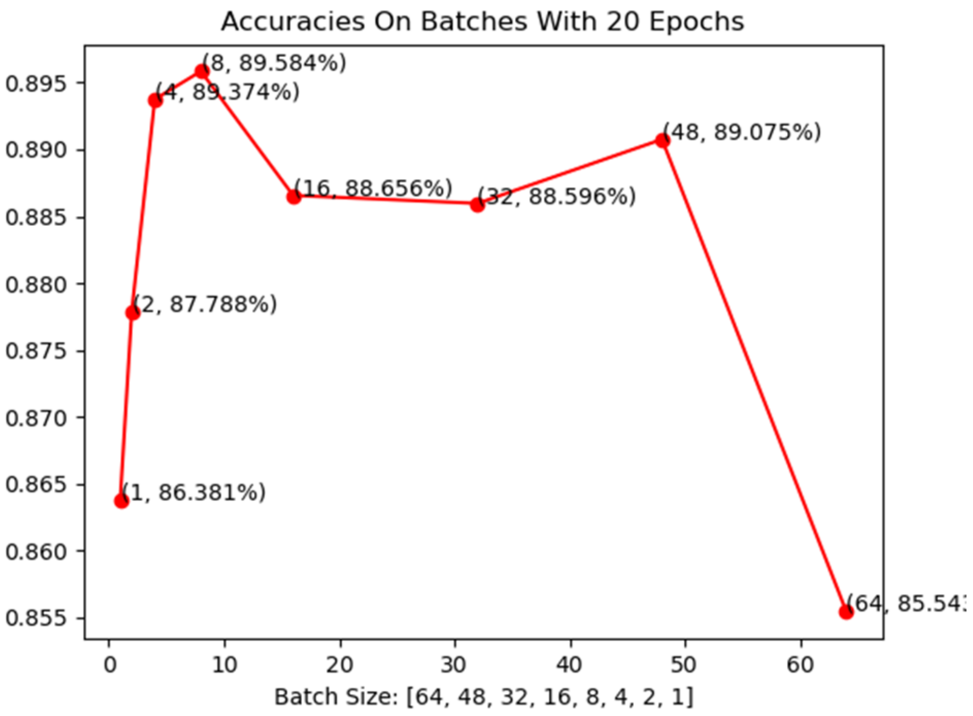


Figure 3.24: Accuracies on all batch-sizes with 20 epochs

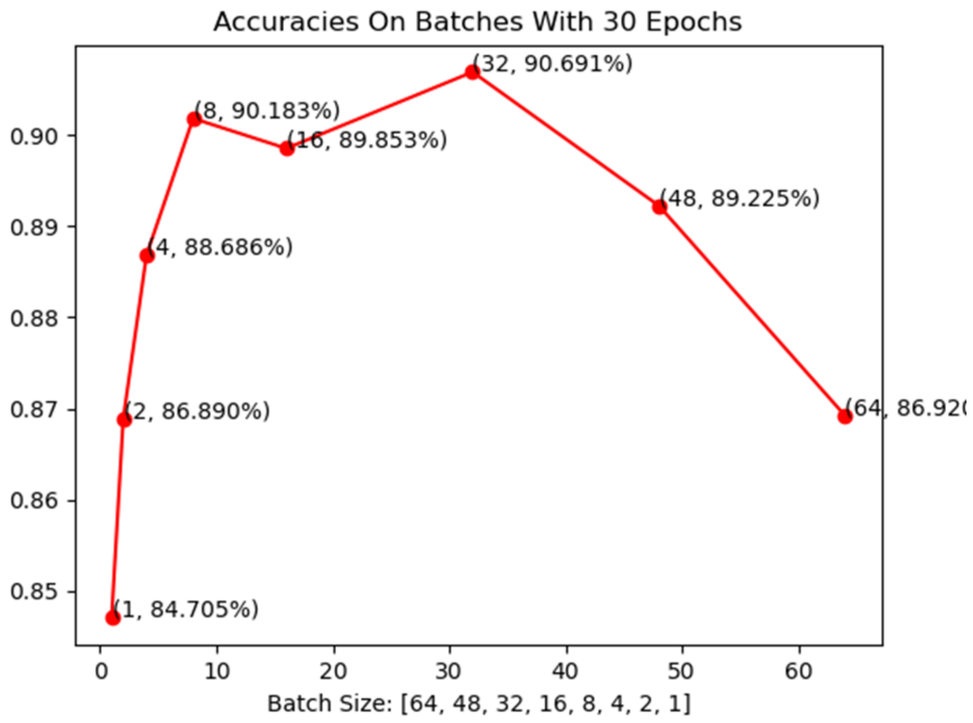


Figure 3.25: Accuracies on all batch-sizes with 30 epochs

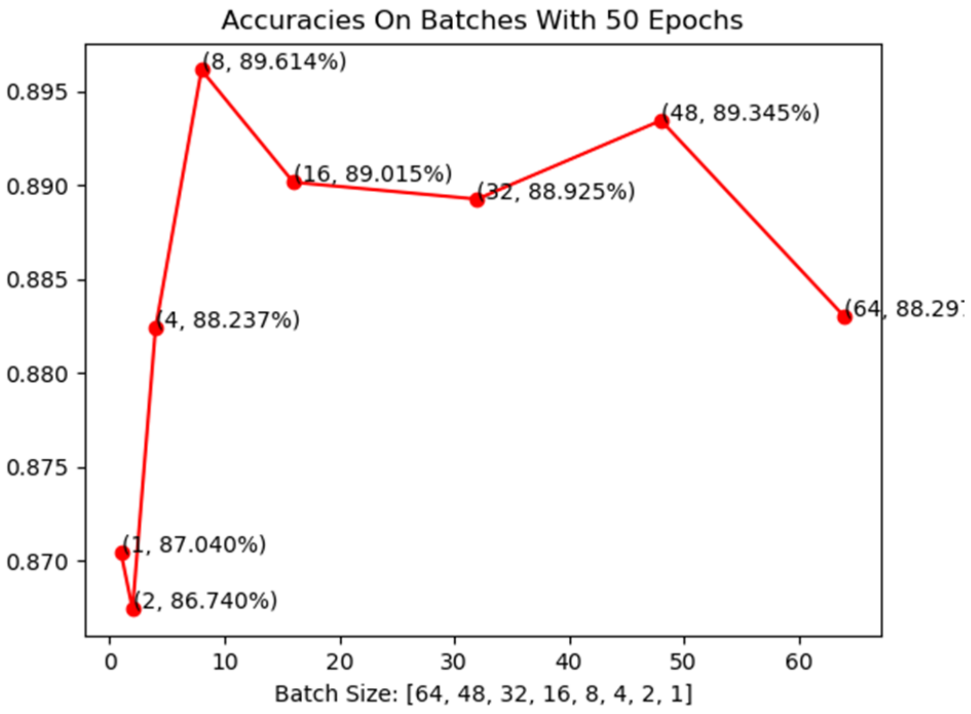


Figure 3.26: Accuracies on all batch-sizes with 50 epochs

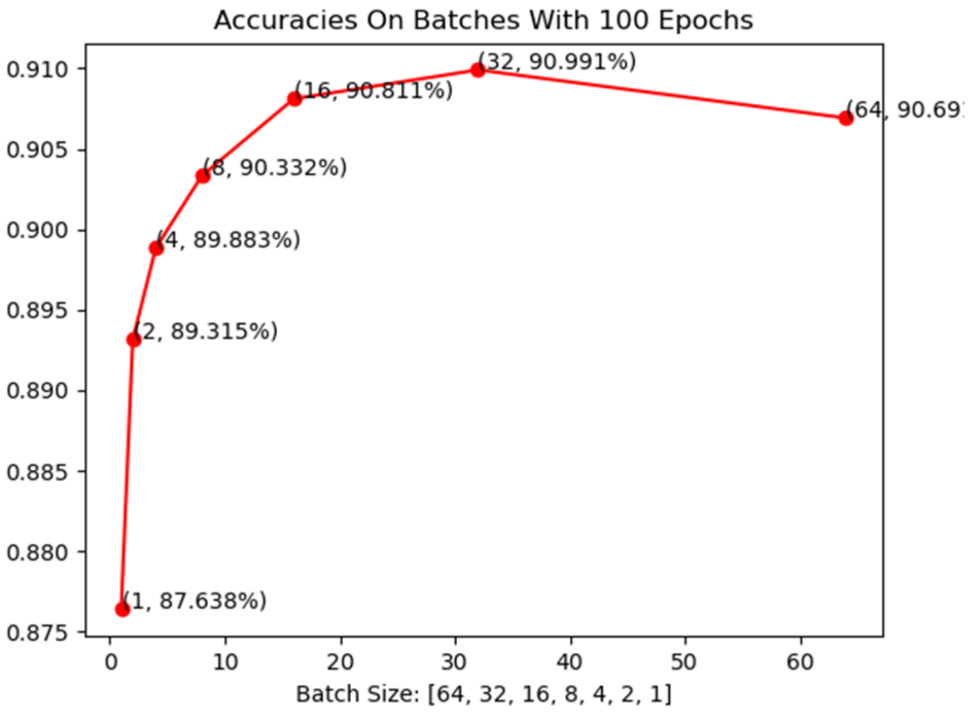


Figure 3.27: Accuracies on all batch-sizes with 100 epochs

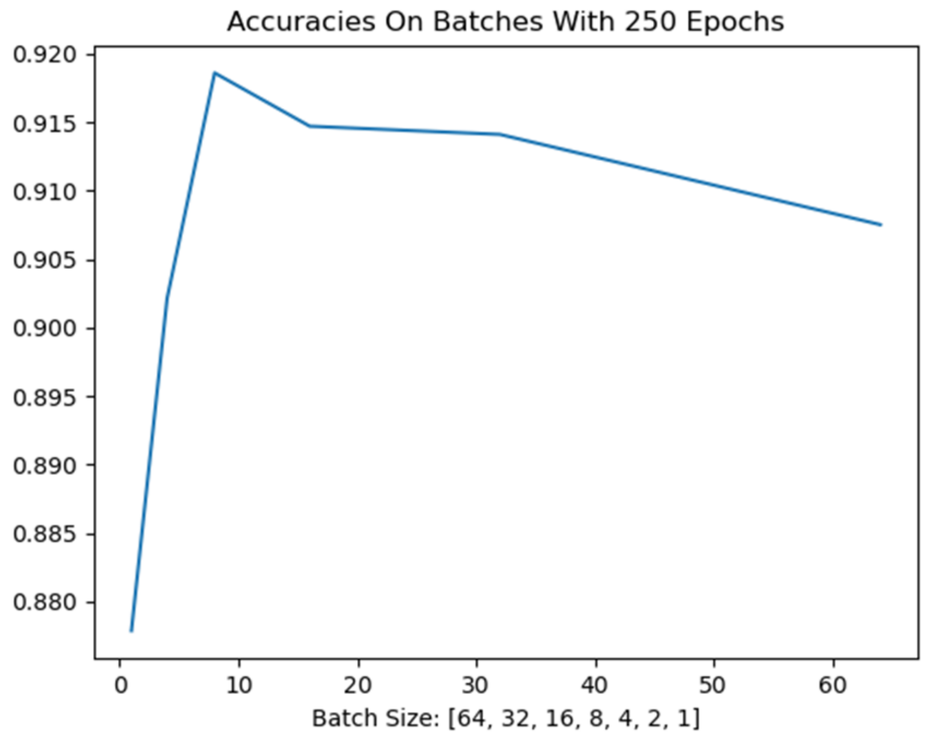
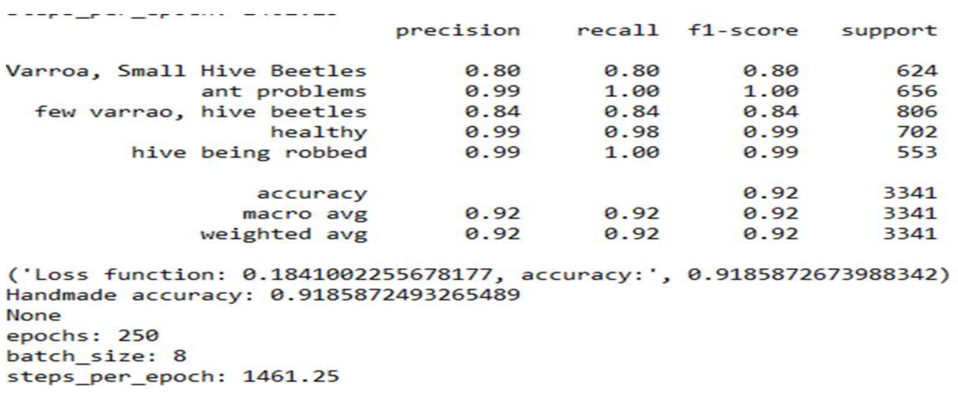


Figure 3.28: Accuracies on all batch-sizes with 250 epochs

The most successful CNN model obtained from this experiment has hyperparameters of 250 epochs and a batch size of 8. The classification report for the most successful model is presented in Figure 3.29, while the confusion matrix is shown in Figure 3.30.



```
                                precision    recall  f1-score   support

Varroa, Small Hive Beetles           0.80      0.80      0.80       624
              ant problems           0.99      1.00      1.00       656
  few varrao, hive beetles           0.84      0.84      0.84       806
                   healthy           0.99      0.98      0.99       702
         hive being robbed           0.99      1.00      0.99       553

                  accuracy                               0.92      3341
                 macro avg           0.92      0.92      0.92      3341
              weighted avg           0.92      0.92      0.92      3341

('Loss function: 0.1841002255678177, accuracy:', 0.9185872673988342)
Handmade accuracy: 0.9185872493265489
None
epochs: 250
batch_size: 8
steps_per_epoch: 1461.25
```

Figure 3.29: Classification report of the most successful model which is 250 epoch, 8 batch-size CNN trained with augmented data

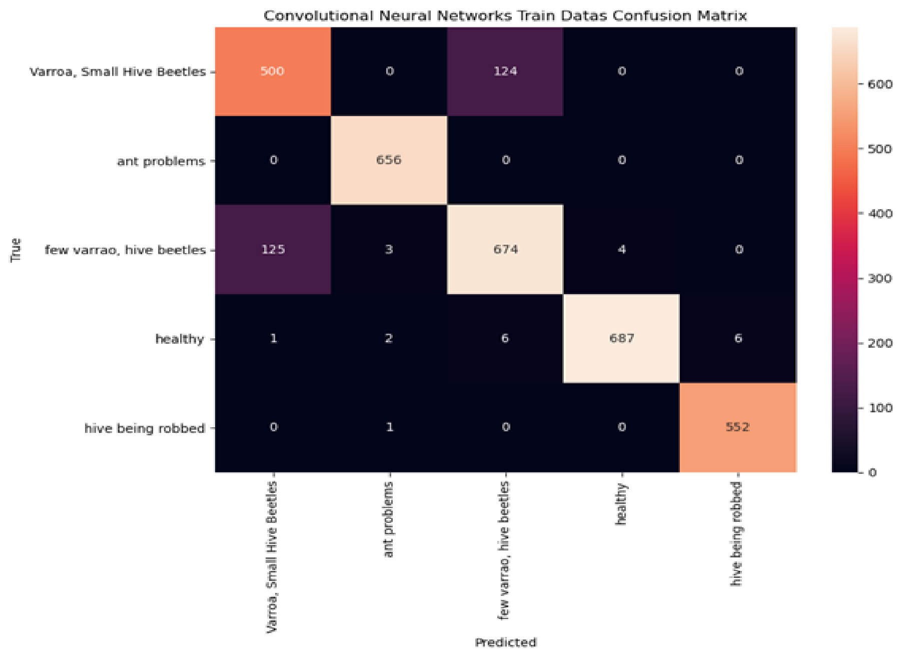


Figure 3.30: Confusion matrix of the most successful model which is 250 epoch, 8 batch-size CNN trained with augmented data

As observed in the confusion matrix, the combination of 250 epochs and a batch size of 8 has resulted in a decrease in the rate of confusion between Varroa classes. Normally, around 30% of the 'few varrao, hive beetles' class would be misclassified as 'varrao, small hive beetles', but with this combination, the rate has dropped below 20%. This improvement can be seen in the classification report, where the overall accuracy has reached 91.8%.

3.6 Object Detection

Our CNN model assumes that each input image represents a single bee for classification purposes. However, in real-life scenarios, the input images provided to the CNN model may not contain any bees at all or may contain multiple bees. In the first case, the model should not make any classification, and in the second case, it should perform separate classifications for each bee in the image, considering their health status. To determine if there are bees in an image and how many bees are present, an object detection model is required. We found a pre-trained model from Roboflow [7] that fulfills this need.

The object detection model from Roboflow provides the center coordinates, height, and width data of all detected bees. Using this information, the corresponding regions are cropped from the original image, formatted as 80x80 RGB images, and provided to the CNN model for health status classification. Based on the health result obtained from the CNN model, a segmentation is performed around each bee, utilizing the center coordinates, height, and width data provided by the object detection model.

In image-based classification problems, segmentation refers to the process of dividing an image into meaningful regions or segments based on specific criteria or attributes.

An example image created by integrating the object detection model obtained from Roboflow and our most successful CNN model, as described above, is shown in Figure 3.31.

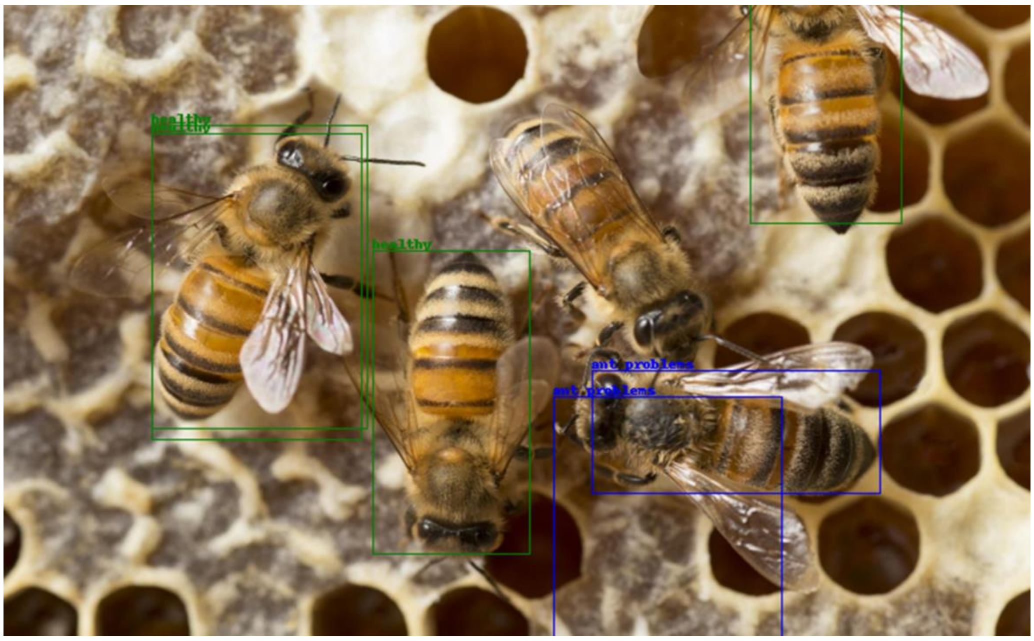
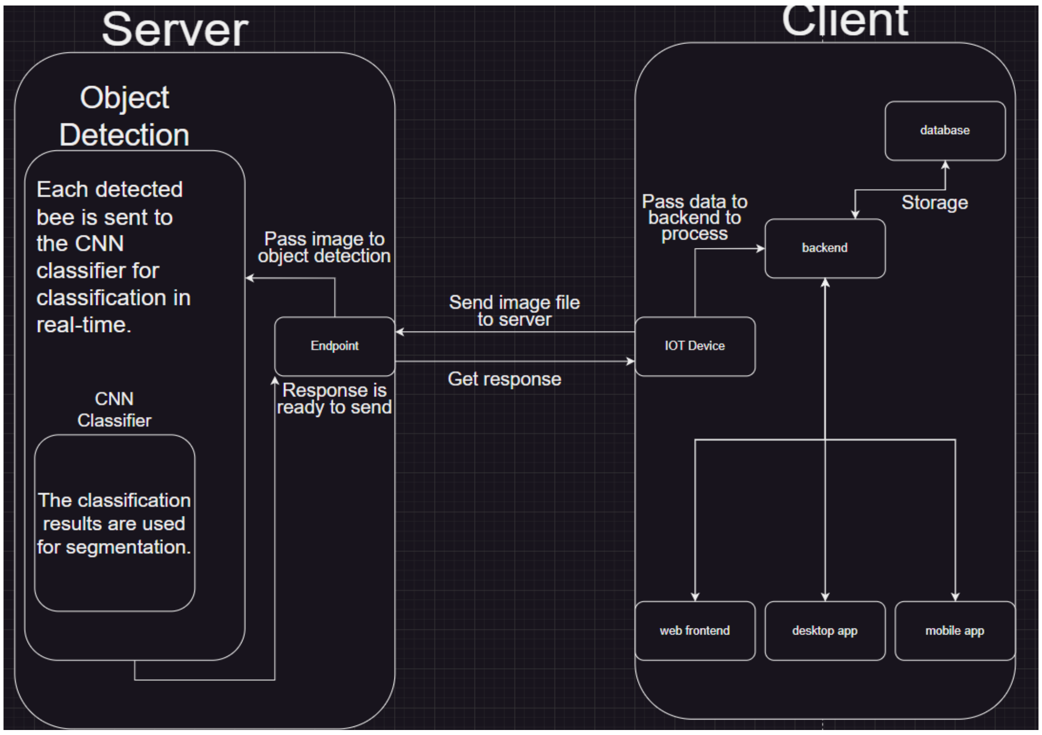


Figure 3.31: An example image generated by integrating the object detection model obtained from Roboflow and our most successful CNN model

4. SYSTEM ARCHITECTURE

The system, in general, receives an image from the client and provides segmented image and some statistical data. The object detection model and the CNN classification model reside on a remote server. The client sends the image file to the server and performs or stores tasks based on the response received from the server. An example system architecture between the client and server is illustrated in Figure 4.1.


Server
Object Detection
Each detected bee is sent to the CNN classifier for classification in real-time.
CNN Classifier
The classification results are used for segmentation.
Pass image to object detection
Endpoint
Response is ready to send
Send image file to server
Get response
Client
database
Storage
Pass data to backend to process
backend
IOT Device
web frontend
desktop app
mobile app
Figure 4.1: Example system architecture between server and client

A mobile application has been developed to provide users with an interactive experience and demonstrate the proper functioning of our system. Through the application, users can send images to the server from the camera or gallery and the application acts as an interface to visualize the response from the server. The server-client architecture for our mobile application is shown in Figure 4.2, the home screen of the application is displayed in Figure 4.3, the image display interface is shown in Figure 4.4, and another image display interface is presented in Figure 4.5.

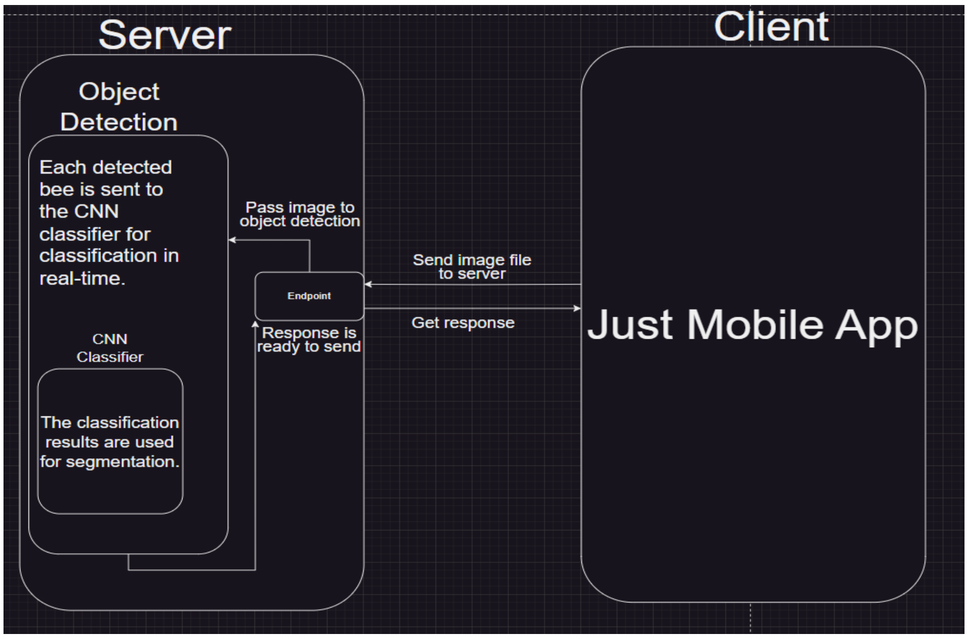

Server
Object Detection
Each detected bee is sent to the CNN classifier for classification in real-time.
CNN Classifier
The classification results are used for segmentation.
Pass image to object detection
Endpoint
Response is ready to send
Send image file to server
Get response
Client
Just Mobile App

Figure 4.2: Server-Client Architecture of our mobile app

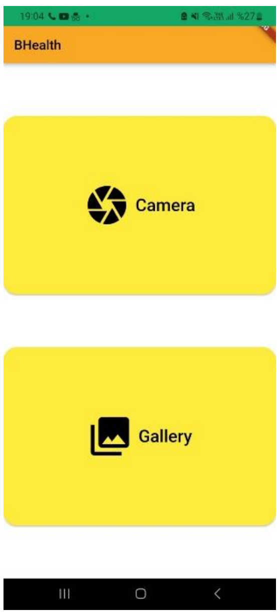
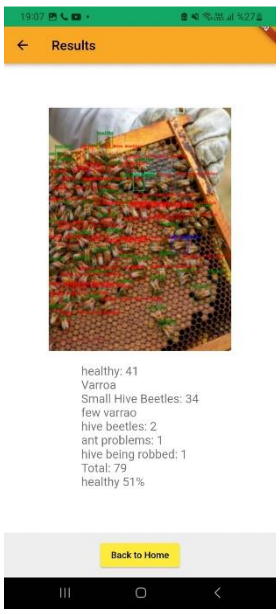


Figure 4.3: Main page



healthy: 41
Varroa
Small Hive Beetles: 34
few varrao
hive beetles: 2
ant problems: 1
hive being robbed: 1
Total: 79
healthy 51%

Figure 4.4: Result screen

X. CONCLUSION AND FUTURE WORK

In conclusion, our project has successfully achieved its’ purpose and has been successfully completed. We have done the required comparisons between different techniques like machine learning and deep learning besides that test if data augmentation has a positive effect on machine learning model. Besides that, a mobile application has been created for an example as the usage of ML model.

In the future works, this model can transform into an IoT device to detect if any problems are occurring in the hive according to health. In addition, this IoT device can be used to acquire massive data about which locations are not good for bees. Moreover, by using this data efficient locations might be found which results in not only cheaper but also easier beekeeping.

REFERENCES

[1] Kaggle. Jenny18, "Honey Bee Annotated Images" dataset. Available: https://www.kaggle.com/datasets/jenny18/honey-bee-annotated-images. Accessed: 1 June 2023.

[2] Python. "Python Programming Language". Available: https://www.python.org/. Accessed: 1 June 2023.

[3] TensorFlow. "TensorFlow". Available: https://www.tensorflow.org/. Accessed: 1 June 2023.

[4] scikit-learn. "scikit-learn". Available: https://scikit-learn.org/stable/. Accessed: 1 June 2023.

[5] Django. "Django Web Framework". Available: https://www.djangoproject.com/. Accessed: 1 June 2023.

[6] Flutter. "Flutter". Available: https://flutter.dev/. Accessed: 1 June 2023.

[7] Roboflow. Matt Nudi, "Honey Bee Detection Model" dataset. Available: https://universe.roboflow.com/matt-nudi/honey-bee-detection-model-zgjnb. Accessed: 1 June 2023.