Individual Report

My focus was leading implementation for the project. Initially he dataset we were given from our other group members was not in the correct format to be passed tino the keras model in order to attempt training, alterations to the format led to us being able to use the data, this simply entailed allocating the train test split of data as numpy arrays rather than regular arrays. This was implemented onto the label arrays in order to change the format from the names of the classes (e.g. n0.. - chihuahua) to integer values v, $v \in \{0 \ldots nClasses\}$. In our initial models this is the implementation that was used however down the line we changed this as our pre-processing had changed completely to leverage image augmentation; this methodology was now not needed. Furthermore I pioneered our group's first implementation of our model in tensorflow's keras. I did this using a manually implemented version of the VGG16 model with no loaded weights, implementing bilinear pooling at the final layers prior to the dense fully connected layer/s. The results from this implementation were promising however validation results were extremely poor in comparison to training results indicating a profound overfitting issue. Finally the last implementation we used was employing pre-trained models from keras and its applications library, these models were trained on the ImageNet dataset and we found Transfer Learning to be the most effective way to achieve high efficacy in our results.

In order to implement these ideas I conducted research into how these tasks are tackled in recent academic papers, a large proportion of these proposed solutions cited the use of bilinear models [3](Lin et al. 2018) in terms of the structure of network layer architecture. Bilinear pooling is implemented for fine-grained image classification, that of the same design as our dataset. Thus this seemed like an appropriate use case based on experimental results according to the literature. This was a result of combined discussion from our literature review group from our project and furthermore a reflection on the results from prior attempts using simple methodologies with Hardik and Kudan from the implementation side of things. We were not satisfied with how the proposed model performed, see Fig. 1.

```
Epoch 59/100
49/49 [==============================] - 73s 2s/step - loss: 0.9580 - accuracy: 0.8899 - val_loss: 2.9178 - val_accuracy: 0.2852
Epoch 60/100
49/49 [==============================] - 73s 2s/step - loss: 0.9424 - accuracy: 0.8919 - val_loss: 2.9168 - val_accuracy: 0.2850
Epoch 61/100
49/49 [==============================] - 73s 2s/step - loss: 0.9238 - accuracy: 0.8971 - val_loss: 2.9221 - val_accuracy: 0.2869
```

Fig.1

We decided that we needed to further research how we would achieve better results through either tuning and adjusting layers/parameters of our already developed model, or try to find an alternative which would provide better results. I went to meet Hardik to discuss this and we came to the conclusion that we were striving for better performance in the model and thought an alternative methodology could be found. From here Hardik came to me with an implementation of the Inception V3 model which had achieved an accuracy of 69% on validation. This was by far our highest performer when considering validation accuracy as the key metric. We found that implementing pre-trained models from the keras applications library loaded with pre-trained weights from the ImageNet dataset resulted in much higher efficacy than our prior attempts [1](Chen et al. 2019). After experimental testing and research into various models, I found that the Xception model, similar to the Inception model, performed even better than inception in terms of validation loss, this indicates that there is a smaller distance between the predicted labels and the true labels, even if the accuracy of the two models is similar meaning the number of correct to incorrect predictions is similar for both models this is nevertheless an improvement. We discuss the reasoning behind this in the group report. Evaluation of other models such as DenseNet201, EfficientNetB7 and ResNet152 resulted in unsatisfactory results. The best of which being ResNet resulting in 76% validation accuracy, which was still worse than ~79-80% given by our prior models. Research in the field led me to papers which discussed InceptionResNetV2 as an effective use case in transfer learning [2](Biswas et al. 2021). The reasoning behind our choice in transfer learning was a group decision which was discussed in depth in the group report. The use of InceptionResNetV2 resulted in validation accuracy of ~84-85%, not only this, however every metric of evaluation we had improved. Loss values of ~0.52 were resultant; whereas in the other two models our lowest values were ~0.65 and ~0.86 for the respective models. Furthermore, I was asked by the Evaluation and Error analysis team to get precision recall and f1 scores for the models we developed. These metrics are depreciated in recent versions of keras and therefore I had to manually implement these modules in order to get these results for analysis. These results were heavily aided by our constant communication with the subgroup involved in implementation.

Having taken Computer Vision as a module in my final year of my Bachelors in computer science led to a thorough understanding of images, processing them and their analyses. Furthermore this was a field in which I was very interested, and led me to write my bachelors dissertation in a similar field, titled, “Hyperparameter optimization and architecture evolution in Convolutional Neural Networks for Image Classification”. Having done similar tasks many times within the research leading up to writing this paper left me with a high level of understanding in regards to these systems, or so I had thought at the time. As much as this project had prepared me for this current task, further research only made me reevaluate the scope of my knowledge as I did not have enough knowledge myself about the specific field of fine-grained image classification. I would say my breadth of knowledge has increased drastically,

My team members involved in implementation had never done similar projects and therefore it seemed appropriate to take the leadership role. I gave myself the responsibility of ensuring all members involved in this aspect were well versed in the specific subject matter. When we first started the implementation Hardik and Kudan were not well informed on the topic, so I spent a day with them showing what we are tasked with, how we can do it, the surrounding theory behind the topics and the practical implementations of these ideas. In terms of my own performance it is difficult to evaluate since without the help of my fellow team members the results would have not been the same; however i would say that i put every aspect of knowledge i had at my disposal to two things: 1 - educating and informing my group members, and 2 - implementing and evaluating the current proposed ideas. I do think that my prior experience helped expedite our development and furthermore assist us in achieving a satisfactory result. If I had to do this project again I would have limited my selection of methodologies, as initially I had multiple ideas which I thought would have worked well, however it likely would have complicated development as I was doing a little work for many different ideas, rather than doing a lot of work for one specific idea. Rather I should have reduced the breadth of research and increased my depth of understanding. As a group I think we worked well implementing our ideas, and further were communicative leading to our ideas not lagging behind one another. Meaning we had the same concepts in mind when considering the development, this was useful as it meant we were on the same page when discussing what was currently being implemented. Contrary to this if i had to improve one aspect of our work together as a group i would have to say the amount of information we were giving each other, as much as we were very communicative the issue was we were lacking in terms of how much information we would give each other, this is possibly a side effect of having to do the work online and communicating over web messengers, however i think ensuring high levels of information traversal between groups would have been beneficial.

Initially I thought of this problem as likely solved by common ANN structures, however with development and further research this was not the case as my understanding of the dataset was not deep enough. My first step in self improvement was trying to fully understand the dataset and how classes varied image to image. My thought process at the time was there were only 120 classes so classification should be straight forward, however understanding the similarity of images between classes led me to researching similar projects; furthermore this changed my scope of information from areas which I was familiar with to those which I had room to improve upon. This transition from concepts I was comfortable with to that which I was uncomfortable with led to education. Without this realisation I would have fallen short of my goals (being high performance in the model evaluation) and would have been stuck trying to fix and improve something that, at its core, was not the optimal way to tackle this task given time and knowledge constraints.

References.

[1] - Chen, P. et al. 2019. Semi-Supervised Fine-Grained Image Categorization Using Transfer Learning With Hierarchical Multi-Scale Adversarial Networks. *IEEE Access* 7, pp. 118650-118668. doi: 10.1109/access.2019.2934476.

[2] - Biswas, A. et al. 2021. Recognition of Local Birds using Different CNN Architectures with Transfer Learning. *2021 International Conference on Computer Communication and Informatics (ICCCI)* . doi: 10.1109/iccci50826.2021.9402686.

[3] - Lin, T. et al. 2018. Bilinear Convolutional Neural Networks for Fine-Grained Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(6), pp. 1309-1322. doi: 10.1109/tpami.2017.2723400.