

**CP 468 – Artificial Intelligence**

# **Spring 2024**

# **Group 25**

# **Project Report**

Member 1 Name: Matteo Passalent ID: 210597410

Member 2 Name: Geon Woo Park ID: 169031088

Member 3 Name: Zaid Rafiq ID: 200772150

Submission Date: August 1st 2024

All the listed members have contributed, read, and approved this submission.

Member 1 Name: Matteo Passalent Signature: Matteo Passalent

Member 2 Name:Geon Woo Park Signature: Geon Woo Park

Member 3 Name: Zaid Rafiq Signature: Zaid Rafiq

**Using a CNN to Identify Lung Cancer from CT Images**

**Abstract**

This report compares the performance of an ensemble model built from three adapted pre-trained models to an image classification CNN made from scratch. All models were trained and evaluated on a small dataset of approximately 1000 CT lung scans. They were assessed on their ability to perform binary classification by identifying each image as positive or negative for lung cancer. The three pre-trained models were developed to build an understanding of the problem and find the optimal configuration for the custom CNN.

**Introduction to the problem**

Lung cancer is the most commonly diagnosed form of cancer at 11.6% in 2018 and is responsible for most cancer-related deaths at 18.4% [1]. Even in recent years, the numbers have continued to increase, as lung cancer takes 12.4% of diagnosed cancers and has a mortality rate of 18.6% in 2022 [2]. This high mortality rate is mainly due to the lack of early detection [3]. If lung cancer could be detected earlier through Computer Aided Diagnostics (CAD), it could significantly lower the mortality rate [3].

CT scanning is the most reliant form of medical imaging compared to X-rays or other methods [4]. Statistics show that early CT screening and proper identification have a significant impact on decreasing mortality rates [5]. Our data includes a sample of CT lung images obtained from a dataset that was created by the Iraq-Oncology Teaching Hospital [7].

**Targets for Performance:**

We set two target metrics to evaluate the final CNN’s performance. Firstly, it needed to identify positive cases of lung cancer with a test accuracy of at least 95%. Additionally, given the limited amount of data, the model needed to be configured to prevent overfitting. To ensure sufficient generalization, we set a target for the difference between the average training accuracy and average validation accuracy (generalization gap), to be no more than 5%.

**Pre-trained CNNs**

We chose three models of varying filter and parameter sizes that would provide good insight into the best configuration for our final model. For each model, we froze its convolutional base and used it for feature extraction, training two new fully connected classification layers on top of it. After the initial training, we unfroze the top block of the pre-trained model to fine-tune it on our dataset. The three models chosen were InceptionV3, VGG16, and MobileNetV2 [8].

InceptionV3 is the largest and most complex of the three with approximately 48 layers and 24 million parameters. Given its size and complexity, we initially expected this model to have the best accuracy. However, due to our small data set and simple binary classification, we found that the larger models were prone to overfitting.

VGG16 is a medium-sized model, it has only 16 layers however each layer has high complexity with around 138 million parameters in total. This model performed similarly to InceptionV3. However, the large amount of parameters appeared to be overly complex for our limited data and resulted in slightly increased overfitting and less overall accuracy. (See Table 1)

MobileNetV2 is the smallest of the three. Although it has around 53 layers, it is a relatively simple model with only 3.4 million parameters. Not only was this model around 3 seconds faster per epoch but it also reached peak accuracy in significantly fewer epochs. This made it much more efficient to train. Additionally, it also had the highest accuracy and the least amount of overfitting. From this, we concluded that a lighter and simpler model was best for our small data set.

While adapting our pre-trained CNNs for optimal performance we established several configurations that improved performance universally across all three models. Firstly, given our limited data set, data augmentation was very effective in improving generalization. Similarly, regularization techniques including L2 regularization and dropout further reduced overfitting to below our 5% target. We found that applying these techniques to the fully connected layer with standard regularization parameters of 0.001 for L2 and 0.5 for dropout was optimal for all three models. Lastly, early stopping was not necessary but helped identify the optimal amount of epochs to run for each model.

**Ensemble Model**

The ensemble model takes an average of each model's prediction and determines a final value. The three pre-trained models use a dense output layer with a sigmoid activation function to produce a probability of lung cancer between 0 and 1. Below 0.5 is classified as positive and above is classified as negative. The ensemble model takes an average of these predictions and returns a final output from all three models. In this process, each model has an equal voting share, which significantly reduces variance and increases reliability.

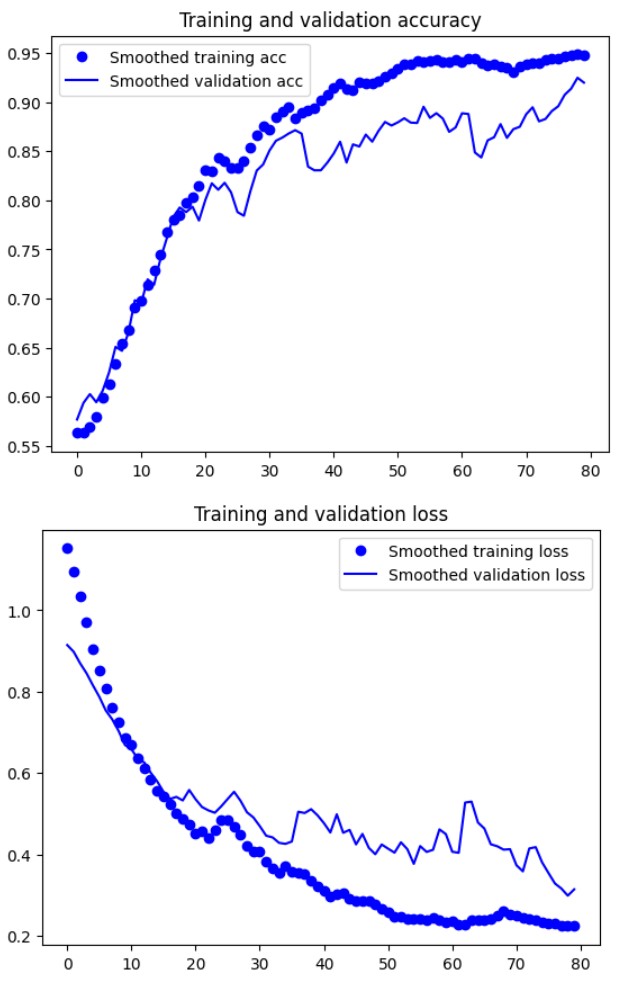
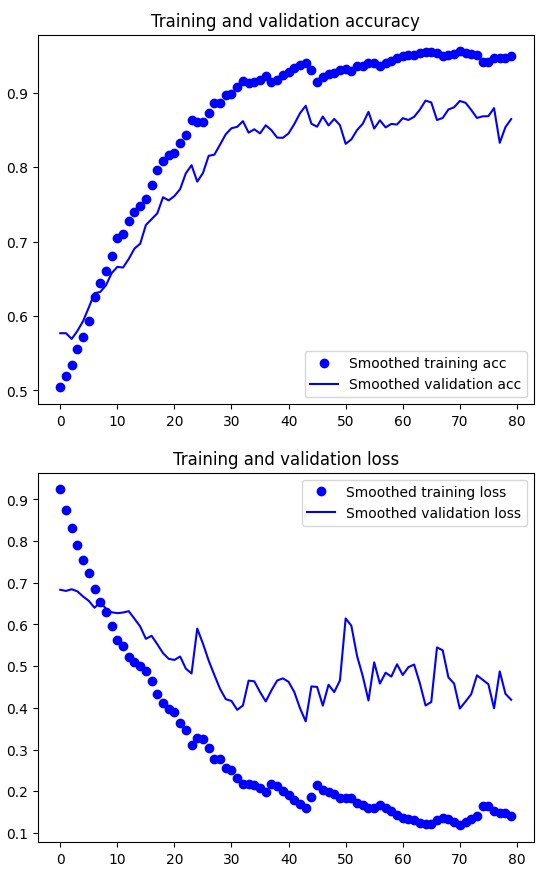
Due to our high accuracy and limited testing data, the final test accuracy for the ensemble model was 100%. Therefore, the confusion matrix and ROC curve are not very informative but are included in the appendix for reference (See Figure 3). However, the improved performance of the ensemble model is evident in its confidence levels when predicting test images. This is because the averaging smooths out predictions and reduces uncertainty, leading to higher confidence levels. When testing the ensemble model through our Gradio GUI a confidence level of 100% is almost always observed.

**Final Custom CNN**

Using the information gained from pre-trained models we built a convolutional model from scratch and fine-tuned it for optimal performance on our dataset. The model architecture was intentionally kept small and simple, consisting of three convolutional layers followed by one fully connected layer. These layers had 16, 32, 64, and 256 nodes, respectively. This configuration proved sufficient for extracting the necessary features to accurately classify images, without overlearning specific features from the training data.

There was some speculation about whether regularization was necessary given our simplified model. We were concerned that since our dense layer had only 256 nodes to begin with, adding L2 regularization and dropout might be contributing to underfitting and preventing the model from learning the necessary patterns in our data. To address this, we did some tests to evaluate the impact of regularization on the model's performance.

The results reaffirmed our findings from the pre-trained models. Not only did the regularization techniques reduce overfitting, but they also improved the overall accuracy of the model on validation and testing data. The graphs below illustrate the enhanced generalization ability of the model with L2 regularization and dropout (right) compared to the same model without regularization techniques (left). The differences in the loss functions are particularly notable, highlighting the significant improvement achieved with regularization.

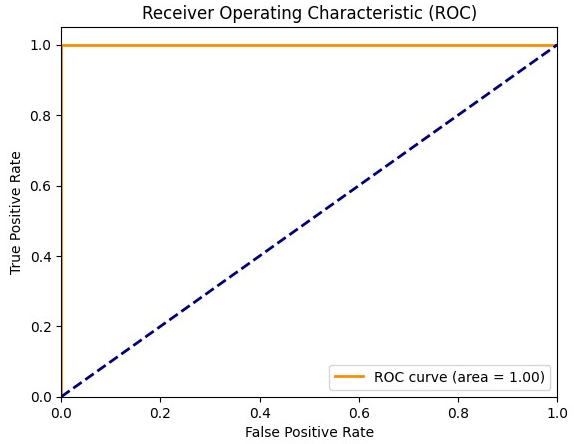
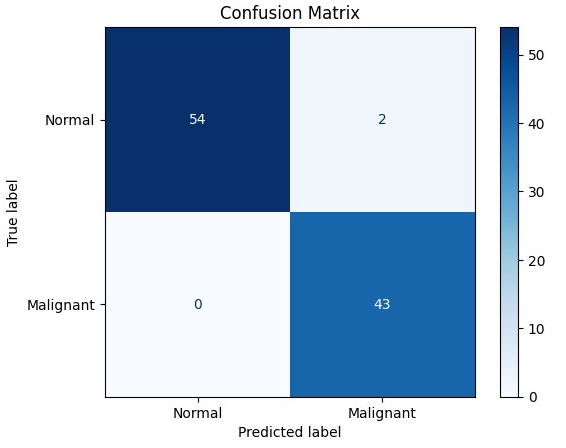


*Figure 1: Loss and accuracy for CNN with no regularization (left) and with L2 and dropout (right)*

After implementing regularization techniques, the generalization gap decreased from 6.6% to 3.9%, comfortably below our target of 5%. Furthermore, the final testing accuracy increased from 81% to 98%, surpassing our target of 95% and highlighting the significant improvement in the model's overall performance.

A deeper analysis of our final model's performance revealed a tendency to favor false positives when inaccuracies occurred. This bias towards false positives is preferable in the practical context of the problem we are solving. The consequences of the model incorrectly classifying a healthy person as positive for cancer are significantly less severe than missing a true positive case. Therefore, the model's bias towards false positives aligns well with the priority of minimizing false negatives at all costs.

The graph below illustrates the confusion matrix and ROC curve of our final CNN evaluated on unseen test data. The confusion matrix shows near-perfect performance with a FN, and FP rate of 0% and 2% respectively. Additionally, the ROC curve, which reflects an AUC of 1.0, also highlights the model's performance. However, the ROC curve is less informative due to the 0% FN rate, resulting in a perfect TP rate and a straight-line representation.



*Figure 2: Confusion matrix and ROC curve of final CNN on test data*

**Additional Discussions:**

Optimizers:

For the three pre-trained models we chose the RMSprop optimizer. This optimizer was used in textbook examples and proved to be best for our pre-trained models. According to both online sources and textbook information, RMSprop is particularly suitable when the learning rate needs to be explicitly adjusted over multiple training epochs. For the pre-trained models, we started with a lower learning rate of 2e-5 for initial training on our dense layers. We then further lowered the learning rate to 1e-5 to fine-tune the top layers of the convolutional base. This lower learning rate helped ensure that the pre-trained weights were not significantly disrupted by large gradient updates.

For our custom CNN, we chose the Adam optimizer. Adam is a modern optimization algorithm that combines the benefits of both RMSprop and the momentum optimizer. It is known for general-purpose effectiveness and performs well without explicitly setting and changing the learning rate.

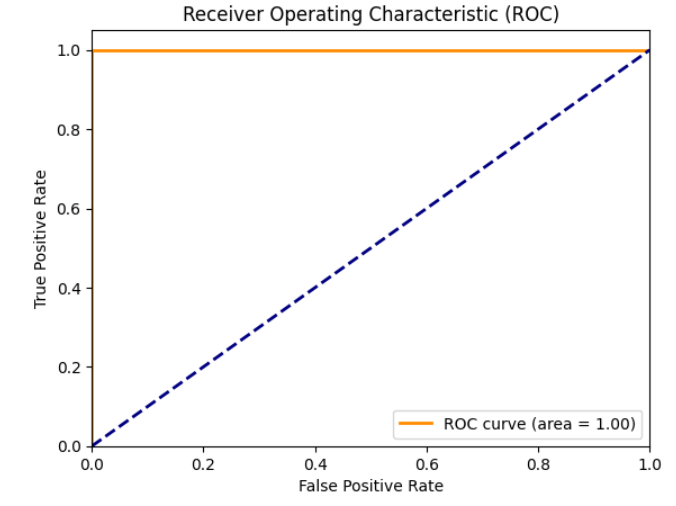
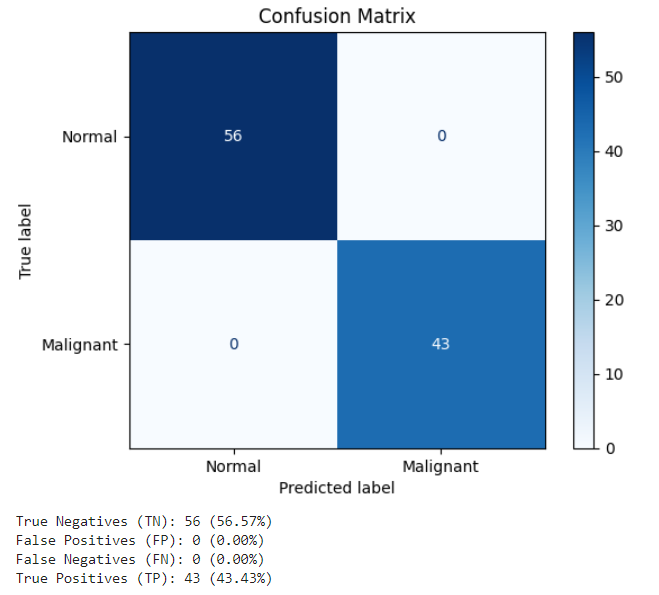
Non-deterministic Nature of Training:  
 During training and evaluation, we observed slight variations in the outcomes each time due to non-deterministic factors such as the random initialization of weights for new layers and the randomly applied data augmentations. In pre-trained models, this variability was minimal because the convolutional base used previously determined weights. However, for our custom CNN every weight was randomly initialized and the differences in outcomes were more pronounced. Most of the time, these discrepancies averaged out, and the final model achieved our target of 95% testing accuracy with a generalization gap below 5%. Occasionally, the model would get stuck in a local maximum early in training, resulting in the validation accuracy remaining at 0.5769 from the first epoch to the 80th. We believe this occurred because if the model did not escape the local maximum early on, the optimizer's adjustments would gradually weaken, causing it to remain stuck indefinitely. This problem occurred rarely and was not a significant blocker for us, we simply restarted the training process when we noticed it was stuck.

**Appendix**

|  | **Model** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Metric** | InceptionV3 | VGG16 | MobileNetV2 | Custom CNN | Ensemble Model |
| Training Acc | 0.97 | 0.95 | 0.97 | 0.88\* | - |
| Validation Acc | 0.93 | 0.90 | 0.95 | 0.84\* | - |
| Generalization Gap | 0.04 | 0.05 | 0.02 | 0.04 | - |
| Testing Acc | 0.99 | 1.00 | 1.00 | 0.98 | 1.00 |

*Table 1: Comparison of training, validation, and testing accuracy for all five models*

\* Note that the training and validation accuracy for the Custom CNN are significantly lower because it was built without leveraging pre-trained weights, resulting in a randomly initialized starting point and low accuracy for early epochs.



*Figure 3: Confusion matrix and ROC curve of Ensemble Model on test data*

**Reference**

[1] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," CA: a cancer journal for clinicians, vol. 68, no. 6, pp. 394-424, 2018, doi: 10.3322/caac.21492.

[2] F. Bray *et al.*, "Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians,* vol. 74, no. 3, pp. 229-263, 2024, doi: 10.3322/caac.21834.

[3] S. Makaju, P. W. C. Prasad, A. Alsadoon, A. K. Singh, and A. Elchouemi, "Lung Cancer Detection using CT Scan Images," Procedia computer science, vol. 125, pp. 107-114, 2018, doi: 10.1016/j.procs.2017.12.016.

[4] A. Al Gindi, T. Attiatalla, and M.-S. Mostafa, “A Comparative Study for Comparing Two Feature Extraction Methods and Two Classifiers in Classification of Early-stage Lung Cancer Diagnosis of chest x-ray images A Comparative Study for Comparing Two Feature Extraction Methods and Two Classifiers in Classification of Early-stage Lung Cancer,” Journal of American Science, vol. 10, no. 6, 2014, Accessed: May 17, 2024. [Online]. Available: https://www.jofamericanscience.org/journals/am-sci/am1006/003\_23291am100614\_13\_22.pdf

[5] Y.-C. Hung, E.-K. Tang, Y.-J. Wu, C.-J. Chang, and F.-Z. Wu, "Impact of low-dose computed tomography for lung cancer screening on lung cancer surgical volume: The urgent need in health workforce education and training," Medicine (Baltimore), vol. 100, no. 32, pp. e26901-e26901, 2021, doi: 10.1097/MD.0000000000026901.

[6] V. Kumar et al., "Risk-Targeted Lung Cancer Screening: A Cost-Effectiveness Analysis," Annals of internal medicine, vol. 168, no. 3, pp. 161-169, 2018, doi: 10.7326/M17-1401.

[7] H. F. Al-Yasriy, “The IQ-OTH/NCCD lung cancer dataset,” Kaggle.com, 2019. https://www.kaggle.com/datasets/hamdallak/the-iqothnccd-lung-cancer-dataset?select=The+IQ-OTHNCCD+lung+cancer+dataset.

[8] Module: tf.keras.applications | TensorFlow Core v2.9.0,” TensorFlow. https://www.tensorflow.org/api\_docs/python/tf/keras/applications

‌