Final Report:
Dermoscopy-Based Melanoma Classification Using CNNs

CS 221 Group Project, Fall 2021

Yifu Chen
yifuchen@stanford.edu

Mariana Irene Frangos
mfrangos@stanford.edu

Jacob Alan Smith
jacobas@stanford.edu

***Abstract*—Melanoma is a type of serious skin cancer that can develop anywhere on the body. Approximately 7,000 deaths are attributed to melanoma each year in the US. The diagnosis of melanoma often requires both a physical exam and a biopsy (removing a skin sample for testing). The cost of biopsies can range from $100 - 300, which is often not covered by insurance and therefore poses a barrier towards early melanoma diagnosis. We plan to develop an image-based melanoma classifier using deep learning, so that people receive quick initial screen for their suspicious skin lesions. We found a suitable dataset on Kaggle, and trained two state-of-the-art Convolutional Neural Network (CNN) models and one custom CNN model to perform this task. We will explore three topics: 1) the benefit of transfer learning from pre-tarined image models, 2) the optimal hyperparameters for fine-tuning these models, and 3) whether CNN models may outperform human dermatologists.**

***Index Terms*—Convolutional Neural Networks, Deep Learning, Computer Vision, Melanoma, Cancer**

I. Introduction

Melanoma is a deadly skin cancer that develops in the skin. At its early stages, the melanoma lesion can be characterized as having asymmetrical brown shaped-mole, with ragged edges along its borders. As melanoma progress, its diameter can reach over 5 mm and penetrate over 4 mm deep in the skin. When melanoma has grown deeply into the subcutaneous tissue, the cancer cells can spread to lymph nodes via the lymphatic system, leading to worsened clinical outcomes.

The traditional diagnosis of melanoma relies on a combination of visual and physical exams. During the initial screening, the physician may ask about the patient's health history and look at the skin lesion to determine if it looks like a melanoma or other types of lesions with similar visual traits (e.g., moles, vascular lesion). The physician may remove a tissue for testing, by using a procedure that uses a circular blade (like a hole-puncher) to cut out the entire lesion. The tissue specimen will be sent to a pathology lab for examination under the microscope. Trained pathologists will dictate the diagnosis to describe the tumor characteristics (thickness of lesion, signs of cancer beyond the skin, proportion of dividing cancer cells etc.).

Determining whether a lesion is a melanoma from just an image is hard (see Fig. 1). Studies show that, when doctors try to diagnose melanoma by only looking at it, the accuracy is around 70% (Esteva et al.). We hypothesize that an artificial intelligence diagnostic tool would at least as accurate as dermatologists, while greatly increase the accessibility and affordability of melanoma screening, such that patients who are concerned about melanoma may use their mobile phone's to screen and decide whether to visit a dermatologist or doctor in person.

We propose to use Convolutional Neural Networks (CNNs) to develop such a tool. Without additional information from biopsies, we must rely on visual features such as the color, size, and shape of the lesion. Since these visual features are able to be captured by RGB images and thus encoded by the CNN architecture using tensors, we will develop and compare various CNN computer vision models for classifying images of the skin lesions. We hypothesize that a binary classification task (melanoma or non-melanoma) is more accurate than a multi-class classification task (melanoma, melanocytic nevus, ...).

To note, our model is not intended to give the final diagnosis, since physical biopsies provide much more information about the skin cell pathologies and are thus the most reliable source of diagnosis.

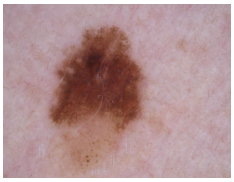
II. Dataset

The input to our model is an RGB image obtained from a dermoscopy. The dataset we will be using contains 25,331 labeled images. Each image may contain a skin lesion that is potentially a case of melanoma. Two examples are shown in Figure 1.

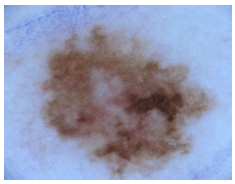
The output is a label that indicates the presence of disease. In the original dataset, each image belongs to one of the following nine categories. Note that "basal cell carcinoma" and "squamous cell carcinoma" both have 99% 5-year survival rate compared to melanoma's 93%, hence we are just interested in detecting melanoma. Therefore we formalized the binary classification task as "malenoma versus all others", while the multi-class task as labeling an image with one of the categories summarized in Table I.

We believe that we can optimize the classification accuracy for melanoma (due to lower approximation error), by grouping 2-9 as a new "non-melanoma" category.

We will be using a subset of this dataset to train our model,



(a) Negative for melanoma



(b) Positive for melanoma

Fig. 1: Two examples from the dataset showing the high difficulty of task. The visual traits of a melanoma lesion is characterized by the ABCD rule (**A**symmetry, irregular **B**order, brown/black**C**olor, $>$ 6 mm in **D**iameter).

Ground Truth Label	Count	Percentage
Melanoma	4,552	17.97%
Melanocytic	12,875	50.83%
Basal Cell Carcinoma	3,323	13.12%
Actinic Keratosis	867	3.42%
Benign Keratosis	2,624	10.36%
Dermatofibroma	239	0.94%
Vascular Lesion	253	1.00%
Squamous Cell Carcinoma	628	2.48%
Total	25,331	

TABLE I: Image ground truth label distribution within the ISIC 2019 dataset. The most frequent labels are Melanocytic, Basal Cell Carcinoma, Melanoma, and Benign Keratosis.

and the rest will be using for validation and testing. In our experiments, we split the training/val/test set to 80%/10%/10%.

III. Literature Review

Using CNN for skin cancer classification is not a new idea. Brinker et al (2018) provided a systematic review of existing work on skin lesion classifiers, concluding that it's difficult to reproduce and compare the performance of CNNs since medical image datasets used by different researchers are rarely available to the public. Therefore, in our paper, we train CNNs under various settings while holding the dataset fixed. Thus we produce a quantitative comparison of CNN architectures, hyperparameters and training methods.

Esteva et al (2017) pioneered the use of CNN for dermatoscopic image classification. They used the GoogLeNet Inception V3 pretrained on ImageNet. Using a dataset set of 129,450 labeled images with 757 skin lesion categories, they achieved an ROC AUC of 0.94 in the melanoma category.

Haenssle et al (2018) also used Inception V3 with transfer learning, but achieved a 0.86 AUC ROC using a dataset of unknown size. Their model outperformed most of the 58 dermatologists (including 30 experts) as well as the top-five algorithms of the ISBI 2016 melanoma classification challenge (Marchetti et al).

While most other researchers kept their model private, Hen et al (2018) released their ResNet model that achieved 0.96 ROC AUC at classifying melanoma. Their model was fine-tuned on 19,398 images. They also performed explainability extraction using heatmaps.

Different CNN ensembles were proposed by Bi et al (2018) and Kawahara et al (2016). The former used combined three different ResNet models to achieve ROC AUC of 0.854 for melanomas. The latter converted one CNN to take multi-resolution input of 227×227 as well as 454×454 versions of the same image, which achieved higher accuracy than their single-resolution setup (0.795 versus 0.781 accuracy). Although both ensemble architectures are novel ideas, we do not explore them in this project due to the lower reported performance than the other simpler models, as well as difficulties in re-implementing the same unreleased architectures.

In addition to ResNet and Inception models, VGGNet has also been used. Lopez et al achieved 78.7% sensitivity by fine-tuning a pre-trained VGG16 network. They found that the fine-tuning approach worked better than using VGGNet as a feature extractor on the test set (although not on the train set).

Researchers also investigated the "training from scratch" approach, although the dataset sizes are an order of magnitude smaller than ours: 3,600 images (Bi et al, 2018) and 136 images (Esfahani et al, 2016). These methods achieved < 0.9 accuracy. Since our dataset has 25,331 images (before augmentation), we believe it's feasible to train our own networks from scratch.

Given the lack of standardized comparison between the existing literature, our paper seems to be the first to conduct a thorough analysis of training melanoma classifiers using CNN. We investigate different settings for the following: 1) image augmentation, 2) CNN model architectures, 3) hyper-parameters, 4) classification task formulation, and 5) transfer learning versus training from scratch. Additionally, we explore whether incorporating image metadata, such as patient's age, sex, anatoimcal site, as input features will improve the model performance.

IV. Methods

A. Baseline

The baseline performance is established by training a simple CNN model from scratch. The model has 5 2D Convolutional layers (kernel size 32, 32, 32, 64, 64 respectively, with "relu" activation) and two Dense layers of size 128 ("relu") and 2 ("softmax"). We also added a and dropout layer to reduce over-fitting. Each layer throughout this CNN are connected by a max-pooling layer. This model has 1,680,834 parameters and we refer to this as **1.68M-Base**. The model summary is shown in Appendix A. We train the baseline model with 5 epochs with Adam optimizer, Cross-Entropy loss, and learning rate 0.0001. The baseline performance are shown in the Results & Analysis section.

B. Main Approach

We use the default implementations and hyperparameters from pre-trained Inception V3 (Szegedy et al, 2015) and ResNet-50 models (He et al, 2015). The training consists of at most 20 epochs (with early stopping) on the original dataset without image augmentation. The classification task is binary, melanoma versus others, because doing so lowers the approximation error and leads to an easier prediction task. We refer to the pre-trained models as as **InceptionV3** and **ResNet-50**. These models used transfer learning from weights pre-trained on ImageNet (Deng et al, 2009). Although it's feasible to train the networks from scratch, due to the size of dataset, we think it will not make much of a difference. We are still trying to figure out how to augment images and prevent disk storage overflow, and leave image augmentation to future works.

We will log the performance of each model from epoch 1 to epoch 20, and analyze the performance with respect to each class (positive for melanoma and negative for melanoma). The performance is done quantitavately through graphs and tables, but also qualitatively by slightly tweaking the hyperparameters and observing the change in perfomance.

When we compare the performance of models trained using different hyperparameters, we select the best hyperparamter based on F-1 score of the melanoma class, and also the overall performance. We perform a hyperparameter search for the baseline, Inception and ResNet CNN models using random-search, which has been shown to be as effective as grid-sweep to find optimal hyperparameters while being more efficient and eco-friendly.

Lastly, to combat the class imbalance in the training dataset, we use a cost matrix that magnifies the loss if the true label is positive. For example, treating a False Negative 10 times as costly as a False Positive would result in the cost ratio of 10:1. We compare different cost matrices to obtain the optimal cost matrix for training the specific model, since different models may have different cost matrices.

C. Evaluation Metric

Melanoma is a life-threatening disease, therefore, for the binary classification task, we are hoping to detect every melanoma case, even at the cost of potentially higher false positives. In the context of our model, a false negative is a image of melanoma that is incorrectly classified as a picture that does not show melanoma. Since our goal is to correctly identify melanoma in patients, it is imperative that we minimize the number of false negatives so as not to cause a delay in patients getting the treatment they need for this very serious disease, assuming this model is applied to real-world cases. Therefore, we choose the main evaluation metrics to be Accuracy and Recall (Sensitivity). Given a confusion matrix:

		True Diagnosis	
		Positive	Negative
Classification	Positive	TP	FP
	Negative	FN	TN

We compute the following evaluation metrics:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}, \text{Recall} = \frac{TP}{TP+FN},$$

$$\text{Precision} = \frac{TP}{TP+FP}, \text{F-1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}.$$

V. Results & Analysis

For initial exploration, we trained each CNN architecture with their default hyperparameters, the results are shown in Table II. We were surprised by the fact that our custom **1.68M-Base** has out-performed the **InceptionV3** model, even though the latter has 24M parameters. The **ResNet-50** showed the most promising performance of 0.859 accuracy at the 5^{th} epoch. However, there is a catch: when we stopped training all models at the 5^{th} epoch, we found that all models are highly biased towards classifying every photos as non-melanoma, which consists the majority $\frac{4}{5}$ of the training data.

Model	Accuracy
1.68M-Base	0.846
InceptionV3	0.829
ResNet-50	0.859

TABLE II: Comparison of baseline model performance with the tuned transfer learning models, using the default hyperparamters.

Although on the surface, it seems that **1.68M-Base** outperformed **InceptionV3**, we noticed that all models performed badly on the melanoma class ($< 5\%$), and are heavily biased towards classifying everything into the non-melanoma class, which consists of the majority of the training set. For example, we noticed that **InceptionV3** tends to classify everything into the non-melanoma class as shown in table III.

Metric	Performance
Accuracy (+ and –)	0.83
Precision (+)	0.00
Precision (–)	0.82
Recall (+)	0.00
Recall (–)	1.00
F-1 (+)	0.00
F-1 (–)	0.90

TABLE III: Example performance of **InceptionV3** with default hyperparamters at epoch 5, showing that the deceptive accuracy of 0.83 is meaningless.

After discovering the problem of the low performance for the melanoma class, we used a custom cost matrix during training as shown in Table IV to extra-penalize incorrect classifications if and only if a melanoma photo is classified as non-melanoma.

		Ground Truth	
		Melanoma	**Non-M.**
Prediction	**Melanoma**	0	1
	Non-M.	arg max Accuracy $\{10,7,5,1\}$	0

TABLE IV: The assigned cost weights of a confusion matrix. For example, the choice of 10-to-1 compensates for the fact that the melanoma images are about 25% of the non-melanoma cases, and that we treat melanoma photos with much more importance as non-melanoma photos.

We trained the three models using their default hyper-paramters again, using the cost matrix specified. We found that the performance of the model increased and is more balanced. During training, we evaluate the intermediate model on the unseen test set after each epoch to monitor the metrics. The results for each model is shown in tables below.

Metric	Epoch 1	Ep. 5	Ep. 10	Ep. 15	Ep. 20
Accuracy (+ and –)	0.65	0.44	0.42	0.48	0.65
Precision (+)	0.17	0.18	0.17	0.18	0.19
Precision (–)	0.82	0.83	0.81	0.83	0.83
Recall (+)	0.26	0.63	0.58	0.54	0.31
Recall (–)	0.73	0.39	0.39	0.47	0.72
F-1 (+)	0.21	0.28	0.26	0.27	0.24
F-1 (–)	0.77	0.54	0.53	0.60	0.77

TABLE V: Epoch-level metrics for the custom **1.68M-Base** CNN model, grouped by photos positive (+) and negative (-) for melanoma.

Metric	Epoch 1	Ep. 5	Ep. 10	Ep. 15	Ep. 20
Accuracy (+ and —)	0.76	0.77	0.77	0.77	0.77
Precision (+)	0.17	0.19	0.20	0.22	0.83
Precision (–)	0.82	0.83	0.83	0.83	0.20
Recall (+)	0.09	0.10	0.11	0.11	0.91
Recall (–)	0.91	0.91	0.91	0.91	0.11
F-1 (+)	0.12	0.13	0.14	0.14	0.14
F-1 (–)	0.86	0.86	0.86	0.87	0.87

TABLE VI: Epoch-level metrics for the fine-tuned **InceptionV3** CNN model, grouped by photos positive (+) and negative (-) for melanoma.

Metric	Epoch 1	Ep. 5	Ep. 10	Ep. 15	Ep. 20
Accuracy (+ and –)	0.30	0.59	0.56	0.48	0.43
Precision (+)	0.17	0.18	0.19	0.17	0.18
Precision (–)	0.81	0.83	0.84	0.82	0.83
Recall (+)	0.79	0.38	0.46	0.52	0.63
Recall (–)	0.19	0.64	0.58	0.47	0.39
F-1 (+)	0.28	0.25	0.27	0.26	0.28
F-1 (–)	0.31	0.72	0.68	0.60	0.53

TABLE VII: Epoch-level metrics for the fine-tuned **ResNet-50** CNN model, grouped by photos positive (+) and negative (-) for melanoma.

To investigate the benefit of transfer learning, in addition to the per-model results shown in the tables above, we compared the performance of **ResNet-50** and **1.68M-Base** in the figure below.

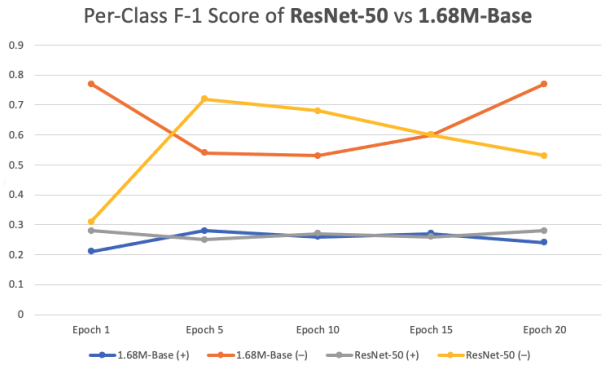


Fig. 2: Performance comparison between the fine-tuned **ResNet-50** and our custom **1.68M-Base** model.

First, we noticed that the imbalance of dataset has negatively affected the models' ability to perform well in detecting melanoma, resulting in a low Recall and F-1 for the photos with melanoma. We tackled this issue by assigning a greater class weight to photos with melanoma than a class weight of 1 to those without melanoma – which forces the model to put more priority on having good recall on the melanoma cases. However, although the model is able to classify melanoma images we found that it came at a sacrifice of the performance on non-melanoma photos. We plan to use several strategies in future work, which includes: using image augmentation, try different learning rates, and tuning the models on more data.

It is challenging for the untrained human eye to perform tasks such as distinguishing between a case of melanoma and a mole (see Fig. 1. In fact, previous studies have shown that even dermatologists have high error rates (without additional information from, e.g., biopsy pathology). In previous works, researchers found that the dermatologists are 65%-80% accurate in melanoma diagnosis (Esteva, 2017).

During our exploration for the optimal cost matrix ratio, we tried 1:10, 1:7, 1:5, and 1:1. The cost ratio that lead to the best performance across all models was around 1:7 or 1:5. A ratio of 1:10 biases the model towards classifying everything as melanoma, while a ratio of 1:1 biases predictions towards non-melanoma. The cost matrix improved the model's ability to predict melanoma classes instead of classifying everything as non-melanoma. However, we think that the cost matrix is one small step forward, since the metrics failed to reach > 0.80 AUC. We think that the reason for the suboptimal performance could be due to the uneveness of the dataset, i.e., too large variance in the images with hairs, scars, different lighting and color conditions, etc.

VI. Error Analysis

The error analysis of our work consists of examining the mis-classified images as well as the following hyperparameters: Cost Matrix, Learning Rate, and Number of Epochs. First, we found that mis-clasified images tend to have a random distribution with no particularly significant patterns. For example, in a sample of 20 mis-classified images, it showed no consistent patterns in terms of their color, lighting, or lesion appearance.

We believe that the error has resulted in suboptimal training and pre-processing. For one, we did not perform image augmentation which may be a limiting factor to the model's ability to generalize well on the unseen test set. Second, we could have selected and tried more hyperparamters, such as learning rate and cost matrices.

We concluded that our experiments show that the optimal hyperparameters for training the models are yet to be found. We think that the former is caused by our computation constraint, since running a grid-sweep of hyperparamter search can cost many days of compute. There could also be a chance that the either the models are loaded incorrectly and thus failed to train, however, since the code ran without errors, and that the models are responsible to changes in hyperparamters (e.g., cost matrix), we think the issue should be something other than bugs in the code.

We think that transfer-learning from pre-trained models (ResNet, Inception, and etc.) fails to provide much improvement, which is probably due to the great difference between the dermoscopy-based and ImageNet images. For example, while an ImageNet image may contain several objects of different shapes in a scene with vairable depth, demoscopy photos contain no objects seen in Imagenet and is 2-D. The ImageNet dataset was a remarkable feat in CNN research, however, our experiment did not benefit from the models pre-trained on it so far.

VII. Future Work

We identified several areas that would benefit from additional research. First, when we assigned the cost matrix for each class to tackle imbalanced images, we qualitatively decided that the ratio is 10-to-1. However, our results show that such configuration may still be suboptimal – we leave investigating the optimal cost matrix to future work since that involves many times more work than what we have time for.

Another area in which we could improve our model is by training it using image augmentation. This would have allowed our dataset to be more balanced and could have produced better results. The reason we could not experiment with any kind of image augmentation strategy is due to the disk space limit of Kaggle.

An area in which we could expand on our current model is to do multiclass classification, instead of a simple binary classification for positive and negative cases of melanoma. In the dataset we used, we were given images of eight different types of skin conditions, many of which would require medical attention or treatment. A model that is able to classify each of these different skin conditions, could be an even more useful tool in the medical field, as it could help patients with all different types of harmful skin conditions get the treatment they need.

Due to the limitation in online VM disk space (20 GB), we couldn't perform image augmentation since the resulting dataset would be many times more the disk space available. In future work, we hope to explore different image augmentation strategies such as performing a forward-selection precedure of image augmentation techniques, determining the effectiveness of flipping, cropping, shifting, color jittering, noise and rotation – and determine whether our results are consistent with existing findings (Shorten & Khoshgoftaar, 2019).

VIII. Code

Our code is executable on Kaggle Notebook's VM platform. Please visit the link below and follow the instructions to run the notebook and see the results yourself.

Kaggle Notebook (run this directly on Kaggle):
https://www.kaggle.com/kagglehof/cs221-project

Github (same code, but you can use your own dataset):
https://github.com/chen-yifu/Melanoma_Classifier

If you need help with running these notebooks, please contact Charles at yifuchen@stanford.edu .

IX. Acknowledgment

We are thankful for the support and mentorship from the CS 221 CA team, especially to our team mentor, Yuchen Wang.

References

Brinker, T.J. et al. (2018) 'Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review', Journal of Medical Internet Research, 20(10), p. e11936. doi:10.2196/11936.

Deng, J. et al. (2009) 'ImageNet: A large-scale hierarchical image database', in 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

Dermatologist-level classification of skin cancer with deep neural networks — Nature (no date). Available at: https://www-nature-com.stanford.idm.oclc.org/articles/nature21056 (Accessed: 11 November 2021).

Haenssle, H.A. et al. (2018) 'Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists', Annals of Oncology: Official Journal of the European Society for Medical Oncology, 29(8), pp. 1836–1842. doi:10.1093/annonc/mdy166.

He, K. et al. (2015) 'Deep Residual Learning for Image Recognition', arXiv:1512.03385 [cs] [Preprint]. Available at: http://arxiv.org/abs/1512.03385 (Accessed: 11 November 2021).

Kawahara, J. and Hamarneh, G. (2016) 'Multi-resolution-Tract CNN with Hybrid Pretrained and Skin-Lesion Trained Layers', in Wang, L. et al. (eds) Machine Learning in Medical Imaging. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 164–171.

Melanoma detection by analysis of clinical images using convolutional neural network — IEEE Conference Publication — IEEE Xplore (Accessed: 11 November 2021).

Nasr-Esfahani, E. et al. (2016a) 'Melanoma detection by analysis of clinical images using convolutional neural network', in 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1373–1376. doi:10.1109/EMBC.2016.7590963.

Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging challenge: Comparison of the accuracy of computer algorithms to dermatologists for the diagnosis of melanoma from dermoscopic images - Journal of the American Academy of Dermatology (no date a). Available at: https://www.jaad.org/article/S0190-9622(17)32202-8/fulltext (Accessed: 11 November 2021).

Shorten, C. and Khoshgoftaar, T.M. (2019) 'A survey on Image Data Augmentation for Deep Learning', Journal of Big Data, 6(1), p. 60. doi:10.1186/s40537-019-0197-0.

Szegedy, C. et al. (2015) 'Rethinking the Inception Architecture for Computer Vision', arXiv:1512.00567 [cs] [Preprint]. Available at: http://arxiv.org/abs/1512.00567 (Accessed: 11 November 2021).

Layer (type)	Output Shape	Param #
conv2d_76 (Conv2D)	(None, 224, 224, 32)	896
max_pooling2d_63 (MaxPooling	(None, 112, 112, 32)	0
conv2d_77 (Conv2D)	(None, 112, 112, 32)	9248
max_pooling2d_64 (MaxPooling	(None, 56, 56, 32)	0
conv2d_78 (Conv2D)	(None, 56, 56, 32)	9248
max_pooling2d_65 (MaxPooling	(None, 28, 28, 32)	0
conv2d_79 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_66 (MaxPooling	(None, 14, 14, 64)	0
conv2d_80 (Conv2D)	(None, 14, 14, 64)	36928
dropout_17 (Dropout)	(None, 14, 14, 64)	0
flatten_17 (Flatten)	(None, 12544)	0
dense_34 (Dense)	(None, 128)	1605760
dense_35 (Dense)	(None, 2)	258

Total params: 1,680,834
Trainable params: 1,680,834
Non-trainable params: 0

Fig. 3: Model summary of our custom **1.68M-Base** model.