SKIN LESIONS CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORK

Parth Maheshwari

Abstract

Given the success of Deep Convolutional Neural Network in Computer Vision tasks such as image classification, object detection, etc. , DCNN has been applied to many other fields and lays path for new research domains. Recently, by transfer learning, Esteva et al proposed in "*Dermatologist – level classification of Skin Cancer with Deep Neural Networks"* that "CNN achieves performance on par with all tested experts, demonstrating an artificial intelligence capable of classifying skin cancer with a level of competence comparable to dermatologists". Inspired by the work, in this project, I attempt to fine-tune deep convolution neural networks that have succeeded with ImageNet dataset for classifying 7 types of skin lesion using HAM10000, a dataset containing 10000 dermatoscopic images. Fine-tuning the top layers was performed with VGG16, Inception V3, Inception ResNet V2 and Dense Net 201. VGG16 and Inception V3 give very similar results on the test set, 79.64% and 79.94% respectively. Inception ResNet V2 performed 3% better, and Dense Net 201 gives even the best single result for fine-tuning top layers with accuracy of 83.93%. Fine-tuning the whole model was performed with Inception V3 and Dense Net 201. After 20 epochs, while Inception V3 performed a little bit better with validation set, Dense Net 201 gives the best single result for test set with accuracy up to 87.725%. The results of experiments verify the intuition that features learned by pretrained models and the architectures of the DCNNs help learning features for a completely different domain dataset, here is the skin lesions dermatoscopic images dataset. Given the computational time and the test accuracy of fine-tuning the top layers and fine-tuning the whole model, for this particular dataset, I find that it's better to fine-tune the whole pretrained model with fewer epochs and less computational time and achieve better accuracy.

I. Introduction

Deep Learning method employs multiple processing layers to learn hierarchical representation of data. It offers a way to harness a large amount of data with few hands feature engineering. Deep Learning method has made impressive advances and evolvement in Computer Vision in recent years, starting from AlexNet in 2012. The image classification task is a very generalized problem: any problems requiring distinguishing between images of different entities can fall within this category. The special characteristics of Deep CNN is that the first layers usually learn very general and "low-level" features of images, while the very last layers of the network learn the semantics and high-level features. Hence, with fine-tuning, Deep Convolutional Neural Networks trained on for image classification task on one dataset can be reused for others image classification task with different dataset. Consequently, fine-tuning has been used widely in Computer Vision research. By fine-tuning InceptionV3, Esteva et al proposed in "Dermatologist – level classification of Skin Cancer with Deep Neural Networks" that "CNN achieves performance on par with all tested experts, demonstrating an artificial intelligence capable of classifying skin cancer with a level of competence comparable to dermatologists". In [1], Esteva et al were using their own obtained dermatologist – labelled dataset consisting of 129,450 clinical images, including 3374 dermoscopy images. This whole dataset includes 2032 skin diseases, belonging to nine skin diseases partitions. By fine-tuning InceptionV3 on this dataset, Esteva et al got up to 66% accuracy classification on nine classes. In this project, I used HAM10000 dataset obtained by ViDIR Group, Department of Dermatology, Medical University of Vienna [2]. Dermatoscopy is a widely used diagnostic technique that improves the diagnosis of benign and malignant pigmented skin lesions in comparison to examination with the unaided eye. The HAM10000 dataset contains 10015 dermatoscopic images collected over a period of 20 year from two different sites, the Department of Dermatology at the Medical University of

Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia. The dataset includes a representative collection of important diagnostic categories in the realm of pigmented lesions: Actinic keratoses and intraepithelial carcinoma / Bowen's disease, basal cell carcinoma, benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses), dermatofibroma, melanoma, melanocytic nevi and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage). Works of this project are as follow:

- Fine tune DCNNs for 7 types of skin lesions for 10000 dermaoscopic images.
- Compare performance of DCNNs: VGG16, Inception ResNet V2, Inception V3, Dense Net 201. Each DCNNs are fine-tuned from the top layers. Fine-tuning the all layers are performed with Inception V3, Dense Net 201.
- Create an ensemble of Inception V3 and Dense Net 201 with all layers fine-tuned.

II. Approach

1. Explore the Dataset

This step involves exploring the data as well as preparing the data for training neural networks. Seven skin lesions in the dataset are: Melanocytic nevi, melanoma, Benign keratosis-like lesions, Basal cell carcinoma, Actinic keratoses, Vascular lesions, Dermatofibroma. While melanoma is extremely dangerous, and Basal cell carcinoma as well as Actinic keratoses can be cancerous, the rest are benign skin lesions. 8000 examples in the dataset are benign and 2000 examples are malignant. The dataset is also biased toward Melanocytic nevi, which has close to 7000 examples. Hence, in the worst-case scenario, our neural network model should get accuracy more than 60%. Figure 2 displays 5 samples for each skin lesion type. For non-experts, it's very hard to distinguish which type is which.

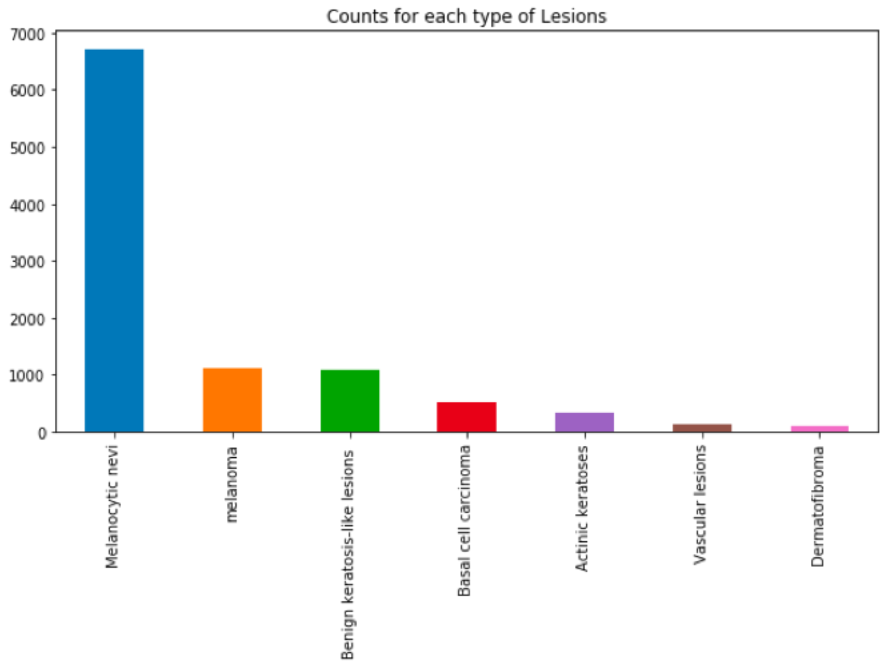


Figure 1

The original images are of size 450 x 600, which is too huge, so I resize the images to 64x64 RGB images for baseline model, and 192x256 for fine-tuning models. The dataset is normalized by dividing by 255 and is split into 7210 training examples, 1803 validation examples, and 1002 test examples.

2. Baseline Model

Before fine-tuning DCNNs, I build a small CNN to estimate the difficulty of classifying skin lesions. The architecture of the CNN is as follow:

- First: A convolutional layer with 16 kernels, each of size 3 and padding such that the size of the image is maintained.

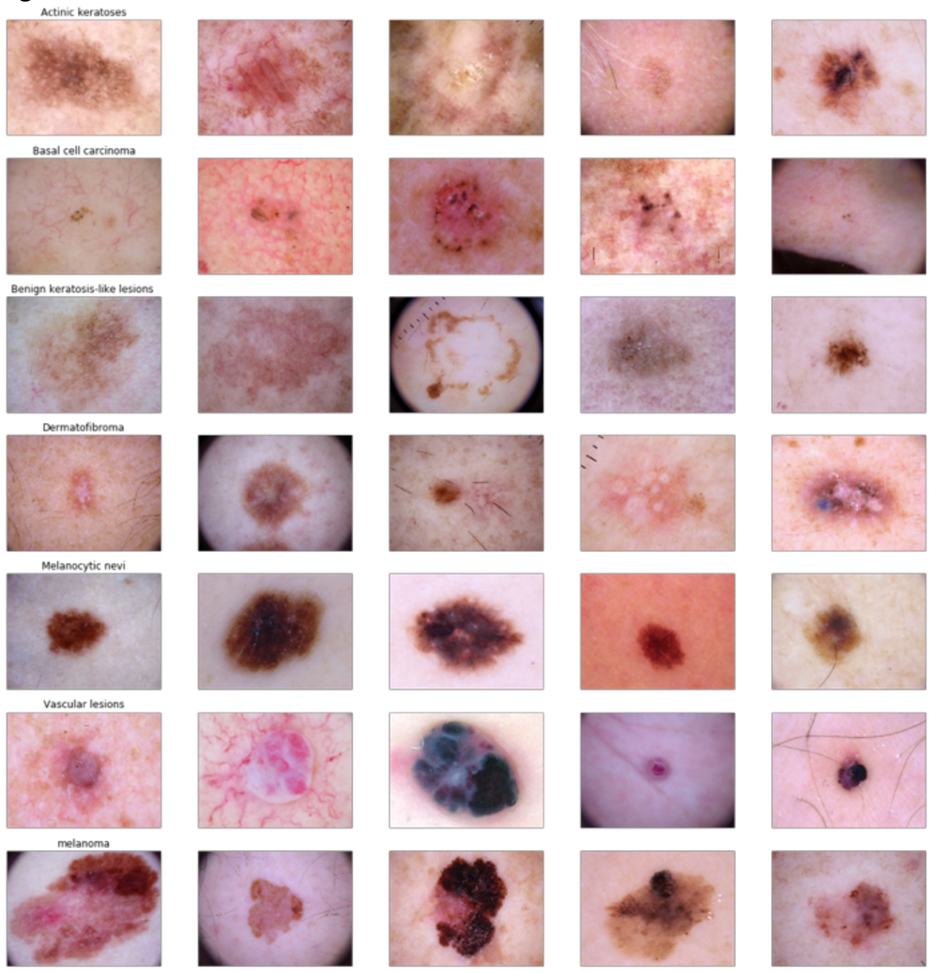


Figure 2: 5 samples for each type of skin lesion

- Second: A max pooling layer with window 2x2. The output is feature maps with spatial activation size reduced by a factor of 2.
- Third: A convolutional layer with 32 kernels, each of size 3 and padding to maintain the same size.
- Fourth: A max pooling layer with window 2x2. The output is feature maps with spatial activation size reduced by a factor of 2.
- Fifth: A convolutional layer with 64 kernels, each of size 3 and padding to maintain the same size.
- Sixth: A max pooling layer with window 2x2. The output is feature maps with spatial activation size reduced by a factor of 2.

The architecture of this model is heuristically based. I follow the convention in famous DCNNs: using the smallest (3x3) convolutional layers; and double the number of filters in the output whenever the spatial activation size is halved to maintain roughly constant hidden dimensions. To train this model, data augmentation is employed. The intuition of this method is to transform the training dataset a bit in each epoch to produce variation and to guarantee that the model will never see the same image twice. Learning rate is initialized at 0.01 and Adam optimizer is used. Learning rate decay is also used so that the learning rate will halve whenever the validation accuracy plateaus for 3 epochs. Baseline model is trained for a total of 35 epochs

3. VGG16

Even though there are many DCNNs model achieving better result on ImageNet than VGG16, I choose to fine-tune VGG16 given its simplicity. Figure 3 is the schema for VGG16 D. The best performing VGG16 net is similar except that the third, fourth and fifth convolutional block has 4 convolutional layers. VGG16 has 0.901 accuracy for top-5 and 0.713 for top-1 on ImageNet.

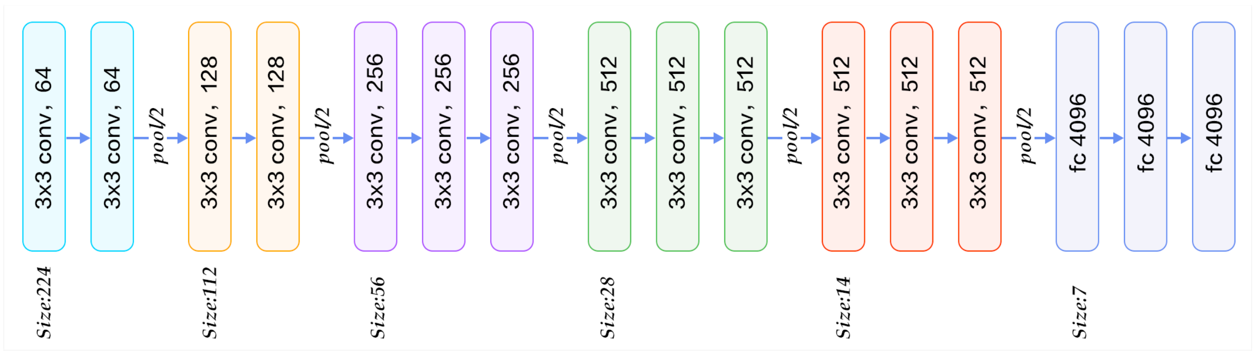


Figure 3: VGG16

To fine-tune VGG16, the top fully-connected layers are removed, and new fully-connected layers (consisting of: one global max pooling layers, one fully connected layer with 512 units, one dropout layer with 0.5 rate, one softmax activation layer for 7 types of skin lesions) for our classification tasks are added. First, freeze all layers in VGG16, and perform feature extraction for the newly added FC layers so that the weights for these layers aren't completely random and the gradient wouldn't be too large when we start fine-tuning. After 3 epochs of feature extraction, we unfreeze the final convolutional block of VGG16 and start fine-tune the model for 20 epochs. Throughout the training process, learning rate of 0.001 and Adam optimizer are used. The same data augmentation and learning rate decay strategy as in baseline model is used. VGG16 was fine-tuned for a total of 30 epochs

4. Inception

Inception V3 was the top performers on ImageNet with 0.937 accuracy for top-5 and 0.779 for top-1. The namesake of Inception v3 is the Inception modules it uses, which are basically mini models inside the bigger model. The inspiration comes from the idea that you need to make a decision as to what type of convolution you want to make at each layer: Do you want a 3×3? Or a 5×5? The idea is that you don't need to know ahead of time if it was better to do, for example, a 3×3 then a 5×5. Instead, just do all the convolutions and let the model pick what's best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions. The larger convolutions are more computationally expensive, so [4] suggests first doing a 1×1 convolution reducing the dimensionality of its feature map, passing the resulting feature map through a ReLU, and then doing the larger convolution (in this case, 5×5 or 3×3). The 1×1 convolution is key because it will be used to reduce the dimensionality of its feature map.

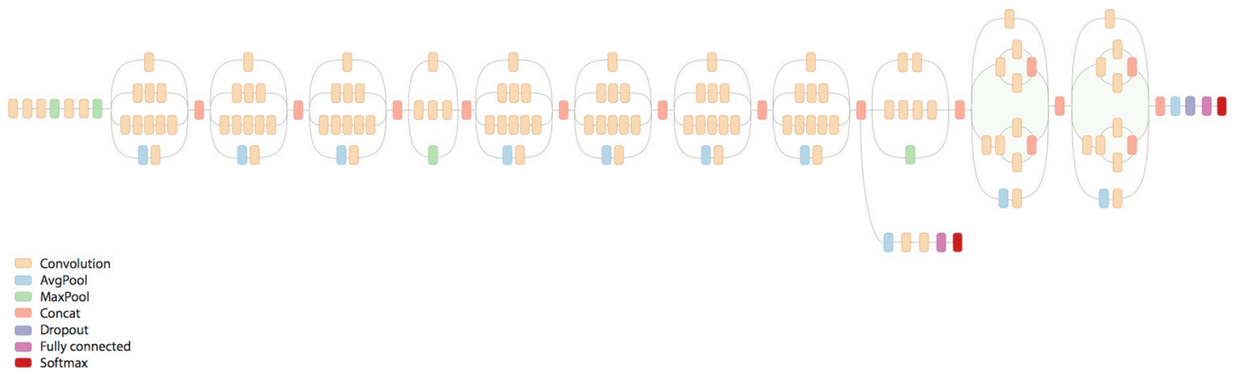


Figure 4: Inception V3 with 11 inception blocks

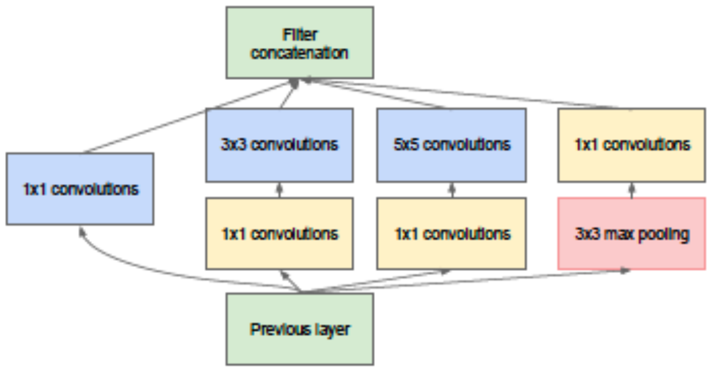
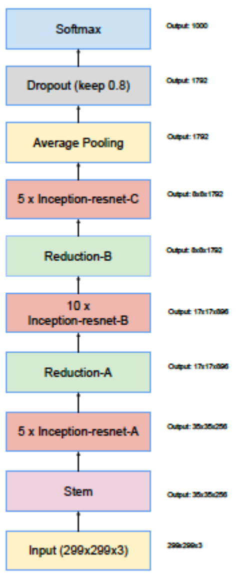


Figure 5: Inception Module

Given Inception V3 trained on ImageNet with 11 inception blocks, I performed 2 kinds of experiment: 1. fine-tuning Inception V3 from the last 2 inception blocks; 2. Fine-tune the whole pretrained model. The reason for this is the implementation of Batch Norm in Keras. The way Keras implemented Batch Norm is as follow. During training the network will always use the mini-batch statistics either the BN layer is frozen or not; also during inference it will use the previously learned statistics of the frozen BN layers. As a result, if we fine-tune the top layers, their weights will be adjusted to the mean/variance of the new dataset. Nevertheless, during inference they will receive data which are scaled differently because the mean/variance of the original ImageNet dataset will be used. Consequently, if I followed exactly what were did when fine-tuning VGG16 with InceptionV3, we will have very bad validation accuracy. This issue was discussed in https://github.com/keras-team/keras/pull/9965 and https://github.com/keras-team/keras/issues/9214. One temporary solution to this issue I can think of is to set all Batch Normalization layer to trainable, so during inference, batch norm layers will statistics of the mini-batch from our training set. I am not sure if this solution is complete, so I decide to fine-tune the all layers of Inception V3 as well as fine-tune the top 2 inception blocks with all batch normalization layers in the model set to trainable. This experiment last for 35 epochs. Fine-tuning the whole Inception V3 was 20 epochs.

Another variant of Inception that is top performer on ImageNet is Inception-ResNet. Inception-ResNet incorporates Residual connection, which is proved to be inherently necessary for training very deep convolutional models. Figure 6 displays the general architecture of this model, and figure 7 displays the architecture of Inception-ResNet-C module, which will be fine-tuned. The same training strategy as in

VGG16 was used for fine-tuning Inception V3, Inception-ResNet V2. Top layers of Inception-ResNet V2 was fine-tuned for 30 epochs.



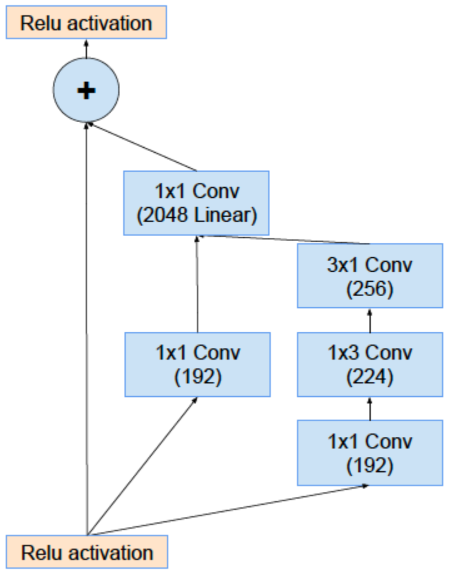


Figure 6: Inception-ResNet

Figure 7: Inception-ResNet-C

5. Dense Net

Dense Net is a new DCNN architecture introduced that is one of the top performers on ImageNet, 0.936 top-1 and 0.773 top-5. The performance of Dense Net is competitive to Inception V3, but Dense Net has less parameters (approximately 20 millions compare with approximate 23 millions of Inception V3). Dense Net 201 has 4 dense blocks, Figure 8 displays the general architecture of a dense block. In a dense block, the l^{th} layer has l inputs, consisting of the feature-maps of all preceding convolutional blocks, and its own feature-maps are passed on to all subsequent layers L – l. Each layer reads the state from its preceding layers and writes to the subsequent layer. It changes the state but also passes on information that needs to be preserved. Dense Net architecture explicitly differentiates between information that is added to the network and information that is preserved by concatenating features instead of summing features as in ResNet.

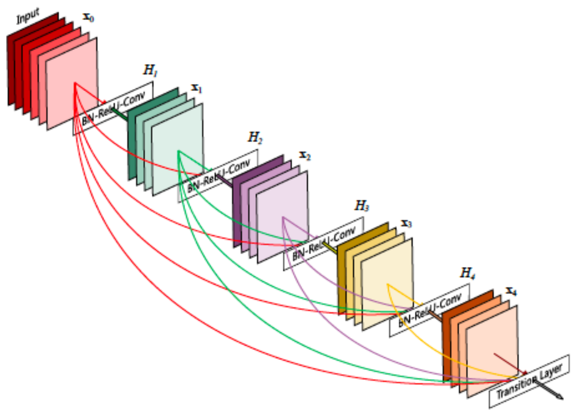


Figure 8: A 5-layer Dense Net block

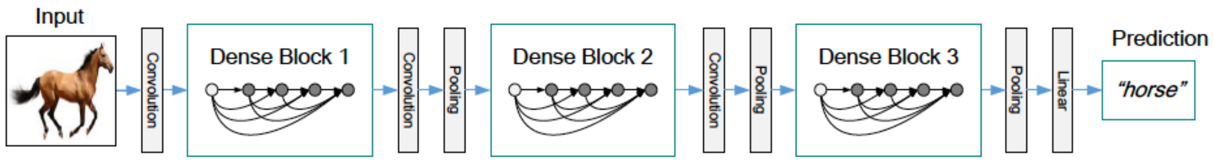


Figure 9: A Dense Net with 3 Dense Blocks

In a dense block, one layer generate feature maps through a composite function, consisting of three consecutive operations: batch normalization, ReLU, and a 3x3 convolution. Convolution and Pooling Layers in the between of each dense block is called together a transition layer. Dense Net 201 consists of 4 dense blocks, and I will perform 2 kind of experiments of this network: 1. Fine-tune on the last dense block (the last dense block has 32 layers); 2. Fine-tune on the whole network. The same training strategy as in previous sections is used to fine-tune Dense Net. Top layers of DenseNet 201 was fine-tuned for a total of 27 epochs, and the whole DenseNet 201 was fine-tuned for 20 epochs. When fine-tuning the whole model of Dense Net 201 and Inception V3, I let the weights to be initialized as the original weights from pretrained model on ImageNet.

III. Results and Analysis

Fine-tuning the top layers

Model	Validation	Test	Test Loss	Depth	# Params
Baseline Model	77.48%	76.54%	0.646671	11 layers	2,124,839
VGG16	79.82%	79.64%	0.708	23 layers	14,980,935
Inception V3	79.935%	79.94%	0.7482	315 layers	22,855,463
Inception ResNet V2	80.82%	82.53%	0.6691	784 layers	55,127,271
DenseNet 201	**85.8%**	**83.9%**	0.691	711 layers	19,309,127

Fine-tuning all layers

Model	Validation	Test	Test Loss
Inception V3	86.92%	86.826%	0.6241
DenseNet 201	86.696%	87.725%	0.5587
Ensemble (Inception V3 and DenseNet 201)	**88.8%**	**88.52%**	**0.41156**

Fine-tuning all layers give us better results than fine-tuning only the top layers, and the time needed for fine-tuning all layers are in fact lower than fine-tuning the top layers. The reason for this is that when fine-tuning all layers, I only performed for 20 epochs, but when fine-tuning the top layers, I perform for 30 epochs. If I only fine-tuned the top layers for less than 30 epochs, the results wouldn't be as good. This observation implies that fine-tuning the whole model not only gives better end result but also helps the model converge faster than fine-tuning the top layers only.

In both cases, Dense Net 201 gives us the best single result, which is amazing given that this model doesn't even have as many parameters as Inception V3. As claimed in [5], Dense Net is a very dense and deep model with just a few parameters. The performance of DenseNet 201 in this experiment verifies the credibility of using DenseNet 201 pretrained on ImageNet for transfer learning in a completely different domain dataset. With ensemble learning, I created an ensemble of Inception V3 and

DenseNet 201, those that were fully fine-tuned previously, and achieved the best result with 88.8% for validation set and 88.52% for test set.

IV. Discussion and Future Work

By the art of transfer learning and ensemble learning, I was able to create an ensemble of fine-tuned Inception V3 and DenseNet 201 and achieved 88.52% accuracy on test set and 88.8% on validation set for HAM10000. Through experiments, I also find that for this dataset, fine-tuning the whole model not only gives better end result but also helps the model converge faster than fine-tuning the top layers only. One serious problem observed during training is overfitting. All of my experiments overfit the training data for 10 – 13%. Many methods are used to minimize overfitting, but I wasn't able to narrow down the amount of overfit further. Future work in avoiding overfitting as well as on better training strategy will help the models to converge to better results.

Reference

[1] Esteva, A., Kuprel, B., Novoa, R., Ko, J., Swetter, S., Blau, H., and Thrun, S. (2017). *Dermatologist-level classification of skin cancer with deep neural networks*. Nature, 542(7639):115–118
[2] P. Tschandl, C. Rosendahl, and H. Kittler, *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*, Sci. Data, vol. 5, p. 180161, 2018
(download: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T)
[3] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition.* In ICLR, 2015.
[4] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. *Going deeper with convolutions*. CoRR, abs/1409.4842, 2014.
[5] G. Huang, Z. Liu, K. Q. Weinberger, and L. Maaten. *Densely connected convolutional networks*. In CVPR, 2017
[6] Kieffer, B., Babaie, M., Kalra, S. & Tizhoosh, H. *Convolutional neural networks for histopathology image classification:Training vs. using pre-trained networks*. arXiv preprint arXiv:1710.05726 (2017)