Insights from HAM10000 Classification Using Learned Compression

Yesh Doctor
Department of Biomedical Engineering
Duke University
Durham, NC 77025
yesh.doctor@duke.edu

Abstract

Neural image compression (NIC) is quickly evolving into a robust and dynamically tunable method for compressing images. In this paper, we sought to investigate how NIC can be used to inform the optimization of inputs for a classification scheme, specifically using the HAM10000 skin lesion dataset. First, we built a classifier on Google's Inception V3 architecture, trained on the data set, and report a test accuracy of 92%. Then, we created a convolutional autoencoder with latent-space dimensions $\frac{2}{3}$rd (1.5X), $\frac{1}{3}$rd (3X), and $\frac{1}{6}$th (6X) the size of the input image, and found that under identical training conditions the 3X autoencoder best preserves image detail relevant for classification. Images were passed through the autoencoder and then the previously trained classifier; all three autoencoders reported similar performance. However, upon jointly training the autoencoders (1.5X, 3X, 6X) with the classifier in a single model architecture, the fully connected model containing the 3X autoencoder performed the best, with a test accuracy of 82%. Upon investigating images before they are passed between the autoencoder and classifier, it appeared that the auto-encoder solely preserved the edges between the skin lesion and its background. Confusion matrix analysis revealed the model performed poorest on lesions with poor lesion-background contrast. This can be used to inform compression schemes that only encode edges, or physical layer optimization to heighten edge contrast before passing into a classifier.

1 Introduction

Modern imaging systems follow a familiar pattern of image detection / collection, digitization, compression, decompression, and finally viewing. However with the adaptation of machine learning based classification, images are not only viewed, but also labeled. Naturally, this raises the question of whether classic image compression (JPEG, PNG, etc.) is optimal for classification schemes, or if so-called "learned" compression methods can eventually become better adapted as compression techniques for images that will be passed into a classifier.

Neural Image Compression (NIC) is one method of learned compression that has recently become a highly feasible compression scheme. The overarching theme of NIC techniques is to design a neural network that can represent an input image in a lower dimension, and provide a scheme to reconstruct the original image from the low-dimensional representation. A specific implementation of NIC is a *comvolutional autoencoder*.

The principal of a convolutional autoencoder is that a high dimensional image, through a series of convolution and pooling operations, can be represented in a low-dimensional "latent space". The convolutional "down" path is called the encoder, and the image representation in the latent space is called the encoding or embedding. The original image can then be reconstructed through a decoder, which is a series of convolutional and upsampling layers that reverse the encoding process

to regenerate the original image. This architecture is typically trained as a unsupervised learning problem where the input data is identical to the target. In the case of a "perfect" autoencoder, the compression will be lossless (loss on all images = 0) and placing a trained autoencoder before a classifier will result in the same test accuracy as passing the test images in alone. However, it may not be necessary for an auto-encoder to be lossless in order to achieve similar test accuracy to the classifier alone. Jointly training the classifier and the auto-encoder on the test set may lead to a more effective compression that represents the input image in a different form after decoding that is more optimized for classification.

We hypothesize that by jointly training an autoencoder and image classifier, we will see better test accuracy than independent optimization. Furthermore, increasing the size of the latent space in the auto-encoder may improve test accuracy since a higher dimensional image representation may preserve more information about the image.

2 Related Work

Li and Ji (2019) [1] describe how neural image compression can be used as a method to explain decisions made by a classifier, and the ideas they present inspired the notion to use an autoencoder to optimize the inputs for a classifier. Esteva et al. (2017) [2] were able to modify the Inception V3 architecture to classify dermatological images to greater accuracy than a board of dermatologists. Sarhan et. al. (2019) [3] have created a variational auto encoder on a dataset similar to HAM10000 that uses adversarial examples to educate the reconstruction of encoded images. Zhgal and Kallel (2020) [4] use an autoencoder as an image preprocessing step to increase melanoma image contrast before passing into a SVM-based classifier. Togacar, Comert, and Ergen (2021) [5] found that autoencoder preprocessing improved the classification ability of MobileNetV2 on malignant skin tumor images. Yap et. al. (2018) [6] found that taking macroscopic images to supplant microscopic skin lesion images may improve classification accuracy. Kompa and Coker (2020) [7] suggest that Unified Disentanglement Networks (UFDN) may be an alternative approach to an autoencoder, as they were able to effectively represent cancer images in a 100-dimensional latent space.

3 Methods

3.1 Data Set

The HAM-10000 dataset of dematoscopic images of common pigmented skin lesions was used. It consists of a 7-way classification task between Actinic keratoses (AK), basal cell carcinoma (BCC), benign keratosis-like lesions (BKL), dermatofibroma (DF), melanoma (MEL), melanocytic nevi (NEV) and vascular lesions (VAS). This dataset was of particular interest as is presents images that could possibly be imaged using a smartphone camera, and hence results may be extrapolated to products for consumer use.

3.2 Classifier Architecture

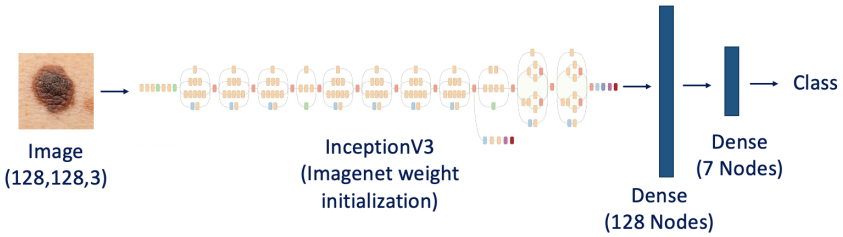


Figure 1: Inception V3 based classifier. (See **link**)

The classifier chosen used a transfer-learning approach on Google's Inception V3 architecture, in line with other transfer learning based classifiers on skin lesions [2].

3.3 Autoencoder Architecture

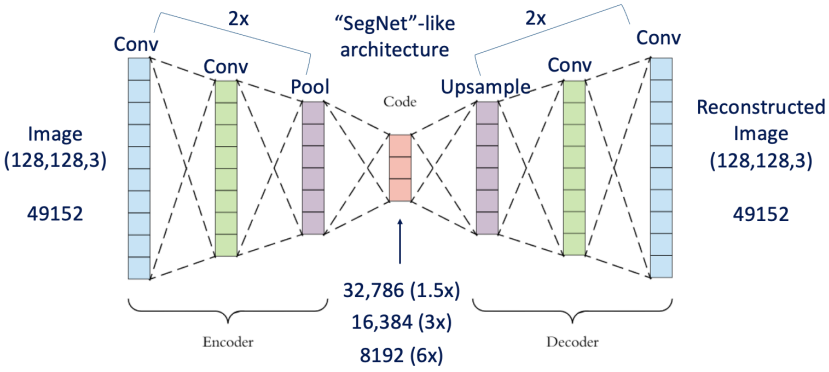


Figure 2: "Seg-Net"-like Autoencoder Architecture (Kudos: Colin)

The autoencoder architecture uses a conv-conv-pool architecture on the encoder side and upsample-conv-conv architecture on the decoder side. The latent space are two convolution layers of with output shape $32 * 32 * n$. n is a variable parameter that is changed to affect the dimension of the latent space. In this case, $n = 8, 16, 32$ was chosen to reflect 6X, 3X, and 1.5X compression of the input image, respectively.

3.4 Full Model Architecture

The input image is simply passed through the autoencoder and then the classifier. Since the autoencoder output shape is identical to its input shape, the autoencoder and classifier are stackable.

3.5 Data Pre-Processing

Input images were resized to 128 x 128, and normalized to have values between 0 and 1. Data augmentation was performed by doing 90 degree rotations, flips, zooms, and shifts.

4 Results

4.1 Classifier Training

The classifier was trained for 35 epochs using the Adam optimizer, categorical crossentropy loss, and l2 regularization on the dense layers. The test loss of 92.3% is on-par with values cited in literature for Inception V3 based classification of HAM10000.

Table 1: Classifier Metrics

Data Set	Loss	Accuracy
Training	0.3819	0.9506
Validation	0.8479	0.8225
Test	0.473367	0.923154

Table 2: Classifier Dataset Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.61	0.69	0.65	110
AK	0.67	0.42	0.52	33
BCC	0.89	0.65	0.75	51
DF	0.70	0.58	0.64.	12
MEL	0.55	0.56	0.56	111
NEV	0.91	0.92	0.92	671
VAS	0.85	0.79	0.81	14
Accuracy			0.82	1002
Macro Average	0.74	0.66	0.69	1002
Weight Average	0.82	0.82	0.82	1002

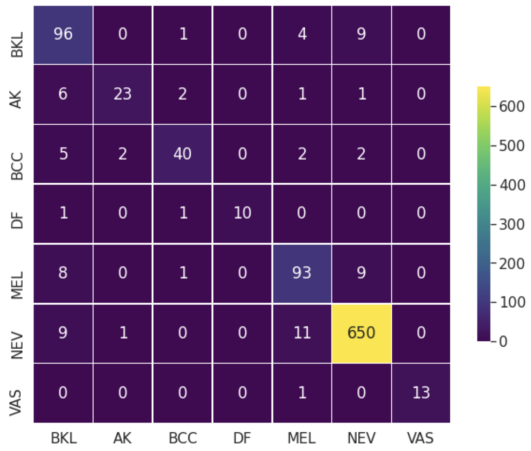


Figure 3: Inception V3 Based Classifier Confusion Matrix

4.2 K-Means Compression

Using a K-Means color-mapping approach, a first naive attempt at compressing HAM10000 images resulted in high fidelity image reconstruction with K as low as 16 (4-bit compression). However, the K-means approach is not jointly trainable with the classifier, and therefore was not used further in compressor generation.

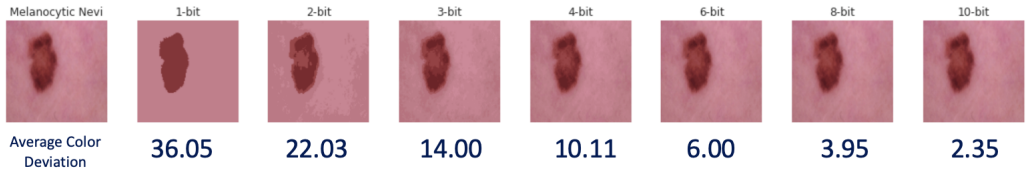


Figure 4: K-means Encoding of Melanocytic Nevi Image

4.3 Autoencoder Training

The autoencoder architecture was trained on the training set, and validated on the validation set using Adam optimization and mean absolute error loss. Training was done in an unsupervised learning manner, wherein the input and target were the same image. Three different autoencoders were trained - ones with a latent-space dimension 1.5X, 3X, and 6X smaller than the input image size. After

training, test images were passed through the autoencoder and then the classifier trained above. Test loss and accuracy were collected for these images.

Table 3: Size of Image Encoding

Autoencoder	Encoding Size
Input Image	49,152
1.5X	32,786
3X	16,384
6X	8192

Data Set	Loss	Accuracy
Training	0.0149	-
Validation	0.0149	-
Test	1.1768	0.7538

Table 4: 6X Autoencoder

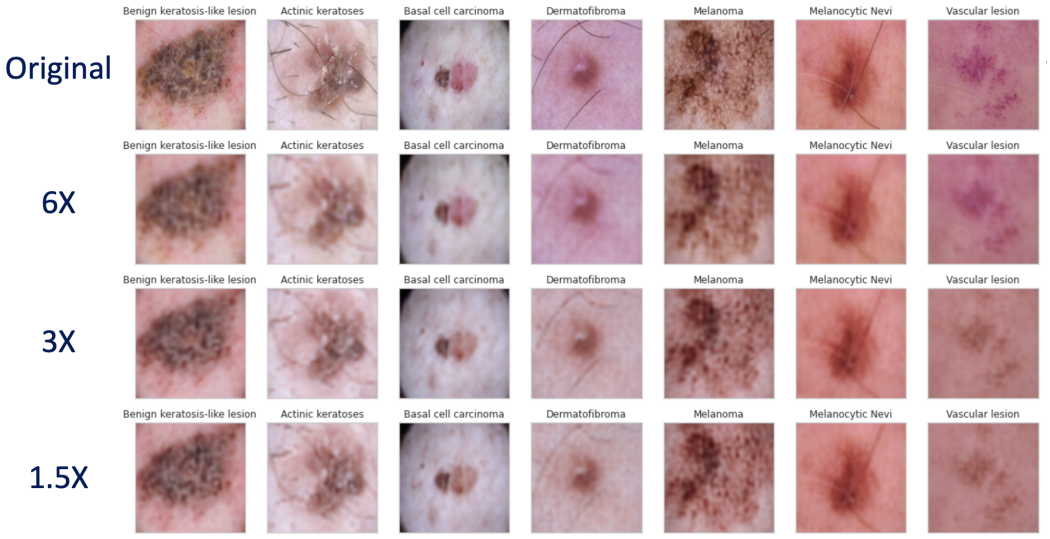
Data Set	Loss	Accuracy
Training	0.0194	-
Validation	0.0194	-
Test	1.2854	0.7315

Table 5: 3X Autoencoder

Data Set	Loss	Accuracy
Training	0.0193	-
Validation	0.0193	-
Test	1.3058	0.7196

Table 6: 1.5X Autoencoder

Figure 5: Sample Test Images Reconstructed Through the Autoencoder



4.4 Joint Autoencoder and Classifier Training

Since the input and output dimensions of the autoencoder are the same, each of the three autoencoder was stacked on top of the classifier architecture, and the entire pipeline was jointly trained using the same training parameters as with the original classifier training. Then, test images were run through the pipeline, and sample test images were extracted before entering the classifier to visualize the new type of image compression the autoencoder is performing.

Data Set	Loss	Accuracy
Training	0.6933	0.7699
Validation	0.7685	0.7432
Test	0.7198	0.7555

Table 7: 6X Joint Model

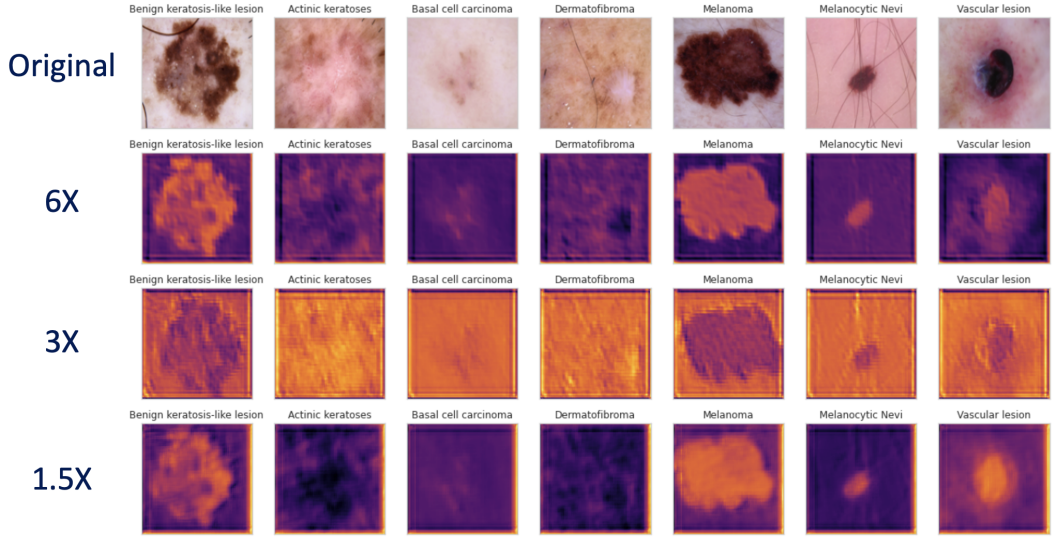
Data Set	Loss	Accuracy
Training	0.5469	0.8261
Validation	0.6289	0.7898
Test	0.5478	0.8183

Table 8: 3X Joint Model

Data Set	Loss	Accuracy
Training	0.6291	0.7869
Validation	0.6870	0.7621
Test	0.6098	0.7814

Table 9: 1.5X Joint Model

Figure 6: Sample Test Images Extracted Before Entering Classifier



5 Discussion

The Incpetion V3 architecture appears to be a robust backbone for HAM10000 architecture. The reported test accuracy of 92% is on par with test accuracy found in literature. [2] Additionally, the training accuracy could likely be improved, as the training loss had not yet plataued after 35 ephocs. From analysis of the precision and recall matrix, it appears that the trained model has difficulty classifying BKL, AK, and MEL. From the confusion matrix, it appears that the classifier thinks that BKL and MEL are NEV samples, and AK samples are BKL samples.

K-means based colormap compression appears to be an effective form of image compression. This method involves extracting the colors from an image, plotting these colors in the "color space", clustering colors into a single color based on the K-means algorithm, and returning a color map where colors are mapped to their single color cluster. The 4-bit compression (16 colors in the colormap) appears to be highly effective, as the reconstructed image looks almost identical to the input image. However, for the purposes of this experiment, it was desirable to jointly train the compressor and classifier to understand how to more effectively compress images for a classifier. Joint optimization of a K-means compressor and a classifier would require redefining a loss function that penalizes larger K. Hence, a convolutional autoencoder was the method that was used going forward.

Surprisingly, the size of the latent space chosen did not appear to make a large difference in compression loss for the convolutional autoencoders. It is possible that the compression is not significant enough for the image loss to be different between a 8192 and 32,786 size image representation. In

further testing, the autoencoder was extended to include 3 conv-conv-pool layers in the encoder (and upsample-conv-conv in the decoder), and performance significantly dropped, to the point that the reconstructed image was completely unrecognizable. Futhermore, passing the decompressed images through the classifier is an unfair test of sorts, as the classifier was not trained on blurry images through data augmentation (this was by design, to see how effective the compression was). The 72-75% accuracy of the autoencoders represents a performance drop of approximately 19% on the test set. Looking at the autoencoder reconstructions of the test images, it appears that the 6X compressor is more color-accurate to the original images, (e.g. no discoloration with DF and VAS), but is more blurry than the 3X and 1.5X compressed images. There is no distinguishable difference between the 3X and 1.5X compressed images.

With the jointly optimized model the 3X compressed autoencoder performs the best on the training, validation, and test sets. This is an unexpected result, but a possible theory could be that in this joint architecture, the 3X model is acting like a segmentation layer, that is segmenting the image to only include the border between the lesion and the background itself - a theory that is backed up by the images extracted before they are passed into the classifier in the joint model. It appears that the images are augmented such that only the boundary between the lesion and the skin is highlighted, whereas most of the other detail is lost. This particular behaviour may be a feature of this data set, since it is searching to classify pigmented lesions on a different colored background. This also explains why the classifier performs poorly on DF and AK (see supplementary figures) as these lesions have limited color contrast with the skin background.

From the joint model, it can be inferred that much of the decision the classifier is making is educated by the shape of the lesion and its contrasting color with the background. This can inform future compression schemes that only aim to encode the boundary between a lesion and the skin, greatly saving on the information that needs to be represented. This can also be used to inform physical layer design; for example, a microscope or camera should be designed to dramatically increase the image contrast with the background, which could improve classification accuracy. Furthermore, a future test could be to arbitrarily increase the image contrast by subtracting the background color and re-running this through the model. This could potentially lead to greater classification accuracy. Finally, to make the autoencoder more task-agnostic, training the autoencoder on multiple data sets may lead to higher compression generalization.

References

[1] Li, Xiang, and Shihao Ji. 2019. "Neural Image Compression and Explanation." arXiv [cs.CV]. arXiv. http://arxiv.org/abs/1908.08988.

[2] Esteva, Andre, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. 2017. "Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks." Nature 542 (7639): 115–18.

[3] Sarhan, Mhd Hasan, Abouzar Eslami, Nassir Navab, and Shadi Albarqouni. 2019. "Learning Interpretable Disentangled Representations Using Adversarial VAEs." arXiv [cs.LG]. arXiv. http://arxiv.org/abs/1904.08491.

[4] Zghal, Nadia Smaoui, and Imene Khanfir Kallel. 2020. "An Effective Approach for the Diagnosis of Melanoma Using the Sparse Auto-Encoder for Features Detection and the SVM for Classification." In 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 1–6.

[5] Toğaçar, Mesut, Zafer Cömert, and Burhan Ergen. 2021. "Intelligent Skin Cancer Detection Applying Autoencoder, MobileNetV2 and Spiking Neural Networks." Chaos, Solitons Fractals 144 (March): 110714.

[6] Yap, Jordan, William Yolland, and Philipp Tschandl. 2018. "Multimodal Skin Lesion Classification Using Deep Learning." Experimental Dermatology 27 (11): 1261–67.

[7] Kompa, Benjamin, and Beau Coker. 2020. "Learning a Latent Space of Highly Multidimensional Cancer Data." Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing 25: 379–90.

6 Supplemental Figures

Table 10: 6X Autoencoder + Pretrained Classifier Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.78	0.26	0.39	110
AK	0.57	0.24	0.34	33
BCC	0.44	0.78	0.57	51
DF	0.00	0.00	0.00	12
MEL	0.52	0.32	0.39	111
NEV	0.81	0.96	0.88	671
VAS	1.00	0.07	0.13	14
Accuracy			0.76	1002
Macro Average	0.59	0.38	0.39	1002
Weight Average	0.74	0.76	0.72	1002

Figure 7: 6X Autoencoder + Pretrained Classifier Classifier Confusion Matrix

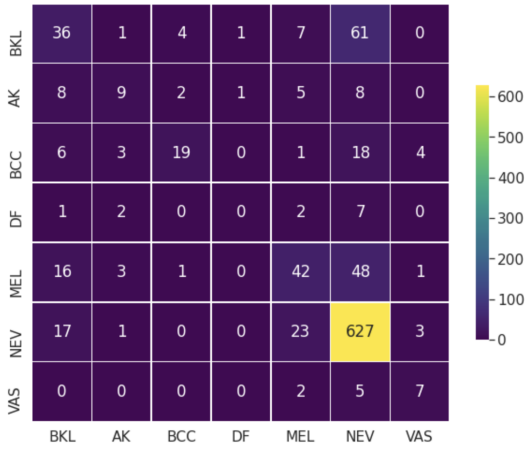


Table 11: 3X Autoencoder + Pretrained Classifier Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.60	0.60	0.60	110
AK	0.67	0.48	0.56	33
BCC	0.63	0.78	0.70	51
DF	0.00	0.00	0.00	12
MEL	0.56	0.48	0.52	111
NEV	0.91	0.94	0.92	671
VAS	1.00	0.93	0.96	14
Accuracy			0.82	1002
Macro Average	0.62	0.60	0.61	1002
Weight Average	0.80	0.82	0.81	1002

Figure 8: 3X Autoencoder + Pretrained Classifier Classifier Confusion Matrix

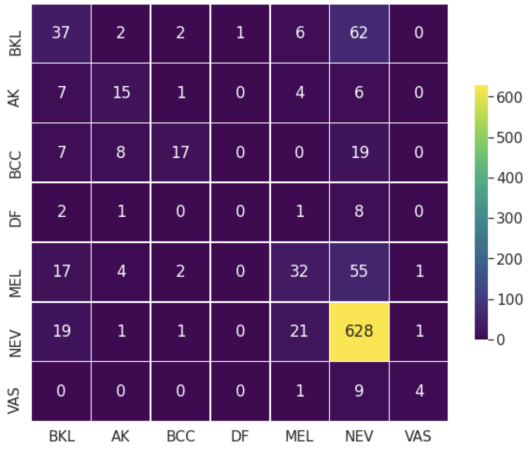


Table 12: 1.5X Autoencoder + Pretrained Classifier Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.60	0.51	0.55	110
AK	0.52	0.33	0.41	33
BCC	0.56	0.61	0.58	51
DF	0.00	0.00	0.00	12
MEL	0.53	0.44	0.48	111
NEV	0.86	0.94	0.90	671
VAS	0.70	0.50	0.58	14
Accuracy			0.78	1002
Macro Average	0.54	0.48	0.50	1002
Weight Average	0.76	0.78	0.77	1002

Figure 9: 1.5X Autoencoder + Pretrained Classifier Confusion Matrix

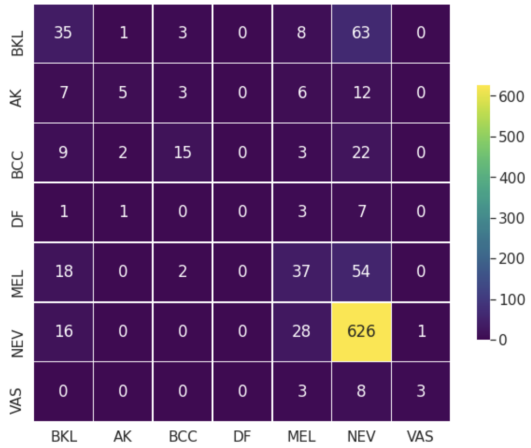


Table 13: 6X Compression Jointly Trained Model Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.78	0.26	0.39	110
AK	0.57	0.24	0.34	33
BCC	0.44	0.78	0.57	51
DF	0.00	0.00	0.00	12
MEL	0.52	0.32	0.39	111
NEV	0.81	0.96	0.88	671
VAS	1.00	0.07	0.13	14
Accuracy			0.76	1002
Macro Average	0.59	0.38	0.39	1002
Weight Average	0.74	0.76	0.72	1002

Figure 10: 6X Compression Jointly Trained Model Classifier Confusion Matrix

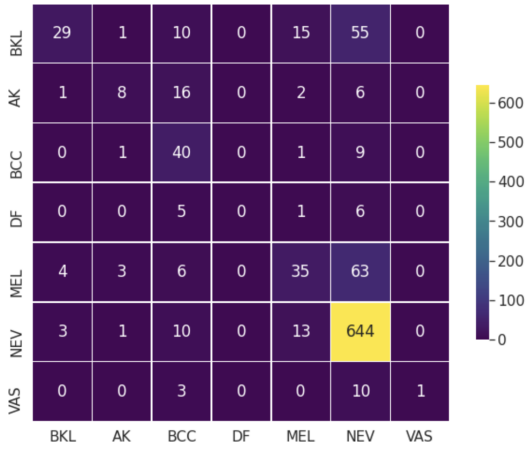


Table 14: 3X Compression Jointly Trained Model Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.60	0.60	0.60	110
AK	0.67	0.48	0.56	33
BCC	0.63	0.78	0.70	51
DF	0.00	0.00	0.00	12
MEL	0.56	0.48	0.52	111
NEV	0.91	0.94	0.92	671
VAS	1.00	0.93	0.96	14
Accuracy			0.82	1002
Macro Average	0.62	0.60	0.61	1002
Weight Average	0.80	0.82	0.81	1002

Figure 11: 3X Compression Jointly Trained Model Classifier Confusion Matrix

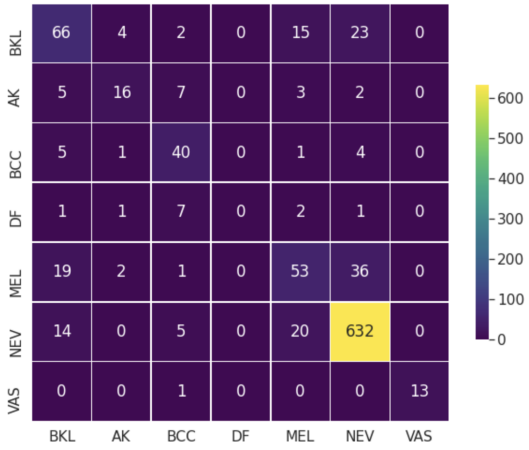


Table 15: 1.5X Compression Jointly Trained Model Precision, Recall, F1-Score, and Support

Skin Lesion	Precision	Recall	F1-Score	Support
BKL	0.60	0.51	0.55	110
AK	0.52	0.33	0.41	33
BCC	0.56	0.61	0.58	51
DF	0.00	0.00	0.00	12
MEL	0.53	0.44	0.48	111
NEV	0.86	0.94	0.90	671
VAS	0.70	0.50	0.58	14
Accuracy			0.78	1002
Macro Average	0.54	0.48	0.50	1002
Weight Average	0.76	0.78	0.77	1002

Figure 12: 1.5X Compression Jointly Trained Model Classifier Confusion Matrix