Effects of Compression on Visual Representations Learned from Medical Images

1st Tomás Barros Everett
Department of Computer Science
Universidad Técnica Federico Santa María
Valparaíso, Chile
tbarroseverett@gmail.com

***Abstract*—Convolutional Neural Networks (CNNs) are widely employed for various tasks in computer vision, and image compresion algorithms such as JPEG have been widely adopted to reduce storage and bandwidth loads. In this paper, we study the effects of JPEG compression on the visual features learned by CNNs by leveraging a hypervector output layer. Results show that a model's learned visual features successfully transfer to compressed images when classifying various skin conditions. The quality of a model also decays in a similar manner to human evaluation of said conditions with extreme compression rates (quality $< 10\%$). These results establish moderate JPEG compression as a viable option at inference time.**

***Index Terms*—Machine Learning, Compression, Hypervectors, t-SNE**

I. Introduction

Convolutional Neural Networks (CNNs) are widely employed for various tasks in computer vision due to their capacity to learn spatial features for visual understanding and decision-making. Compression algorithms, commonly applied to reduce storage requirements, introduce artifacts that may lessen the applicability of said spatial features. We investigate the effects of compression on CNN's final feature representations by leveraging a hypervector output layer which allows the direct application of clustering and visualization techniques to the results of an image classification task. Results indicate that CNNs trained on uncompressed datasets generalize well when performing inference on compressed images even with high degrees of compression. Furthermore, better models construct learned representations which are more robust to compression artifacts.

II. Background

JPEG

For the past 30 years, Joint Photographic Experts Group (JPEG) has been the most widely used image compression standard in the world [3]. As modern image processing tasks require extensive compute times and and storage space, compression algorithms offer a solution to reduce the load of working with large image datasets. Convolutional Neural Networks (CNNs) have long been the standard for image classification tasks [6], and as such, an evaluation of the effect of JPEG's compression artifacts on the network's learned representations is of relevance.

Architecture: HD-MANN

The architecture employed in this study is a Hyperdimensional Memory Augmented Neural Network (HD-MANN), which corresponds to a traditional CNN with a hypervector output instead of the standard one-hot encoded output vector. The explicit memory (EM) of the network is a set of key-value pairs where each key is a high dimensional binary vector (usually between 500 and 10.000 bits long), and each value is a class label. To assign a key to a class label, a random vector is sampled from a discrete uniform distribution $U\{0, 1\}$. Random vectors in high dimensional space are likely to be far away from each other [5], thus providing prediction templates that allow a network to distinguish between classes. Once trained, inference is performed by obtaining the high-dimensional output of the network and searching the EM for the closest known vector.

There are several advantages to working with hypervector outputs:

1) Clustering and visualization techniques can be directly applied to the network's outputs.
2) New classes can be added to a model without having to modify the model's architecture [2].
3) Anomaly detection algorithms can be used to detect out-of-distribution data through distance metrics.
4) Outputs can be used as Joint Embeddings for compatiblity between predictive models.

Though said output representation also produces longer training times proportional to the length of the hypervector, as the amount of gradient calculations performed in the backward passes increases linearly with the amount of output neurons.

III. Experimental Setup

Data

The dataset used for testing comes from The International Skin Imaging Collaboration (ISIC) and is composed of 2357, 180x180 RGB images of 10 skin diseases and conditions. To facilitate illustration, only the top 4 most common classes were used (see Fig. 1), these being:

1) Melanoma
2) Nevus
3) Basal Cell Carcinoma
4) Pigmented Benign Keratosis

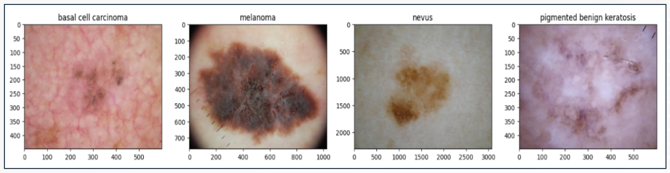

Fig. 1. An example of each of the 4 selected classes. From the left: Basal Cell Carcinoma, Melanoma, Nevus and Pigmented Benign Keratosis

We hypothesise that JPEG compression will have a strong effect on a classification model's performance on this dataset, as the diagnosis of skin conditions greatly relies on border and colour information [4].

Model implementation

The architecture of the HD-MANN is that of a standard CNN, with a hypervector output layer (see Table I)

TABLE I
CNN Architecture Details

Layer	Output Shape	Param #
Rescaling	(None, 180, 180, 3)	0
Conv2D	(None, 180, 180, 64)	1,792
MaxPooling2D	(None, 90, 90, 64)	0
Conv2D	(None, 90, 90, 128)	73,856
MaxPooling2D	(None, 45, 45, 128)	0
Dropout	(None, 45, 45, 128)	0
Flatten	(None, 259,200)	0
Dense	(None, 1,000)	259,201,000
Dense	(None, 500)	500,500

Data augmentation was used to improve the model's generalization capability. Probabilistic rotations, zooming and flips were applied to obtain 600 samples per class from the original 400.

The CNN was trained using AdamW [7] as the optimizer and Mean Squared Error (MSE) as the loss function. While common knowledge indicates the use of Categorical Cross Entropy for binary output vectors, the lack of a final softmax layer greatly impedes learning. A train/validation split of 80/20 was used.

The model was trained on the full-quality image set for 200 epochs with early stopping, resulting in 90 effective epochs. To evaluate whether learned features generalize to JPEG compressed images, we produce 6 parallel datasets compressed to 50,20,10,5,3 and 1% quality levels (see Fig. 2).

Results visualizations with t-SNE

t-SNE [8] is a visualization algorithm that iteratively learns a low-dimensional embedding of high-dimensional datapoints while maintaining the relative distances between them. We apply t-SNE to each dataset's predictions to visualize the distinguishability of the model's outputs as a function of the compression ratio (see Fig 3). It is worth noting that while the distribution of datapoints may vary widely between t-SNE executions, what matters is the distance between datapoints, as this is what t-SNE maintains in the low dimensional embedding of the hypervector space.

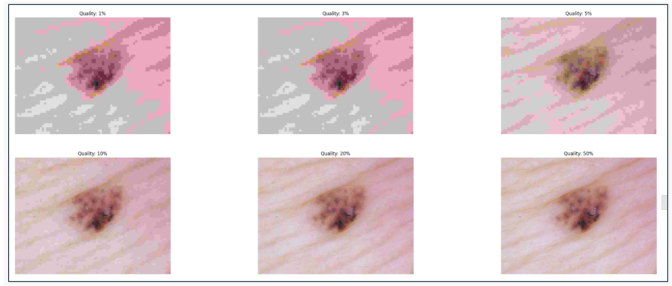

Fig. 2. An example image compressed to various degrees of remaining quality. In order from left to right, top to bottom: 1%, 3%, 5%, 10%, 20% and 50%

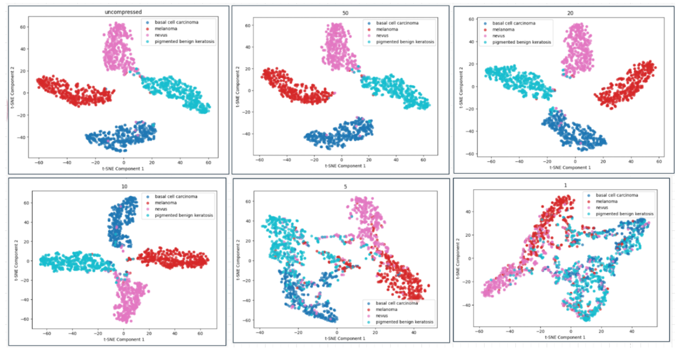

Fig. 3. t-SNE applied to the model's high-dimensional outputs. The model produces a clear clustering of datapoints per class when fed uncompressed inputs (see first image, top left), however clustering quality greatly diminishes when compressing to 5% quality levels.

With a properly trained model using high-quality images, each class's predicted datapoints cluster around it's template hypervector. As the compression quality lessens, the clusters lose density and join together. When class clusters begin to overlap, we can expect the network to be incapable of correct classification. Consequently, we see a drop in accuracy proportional to the amount of compression applied (see Fig. 4).

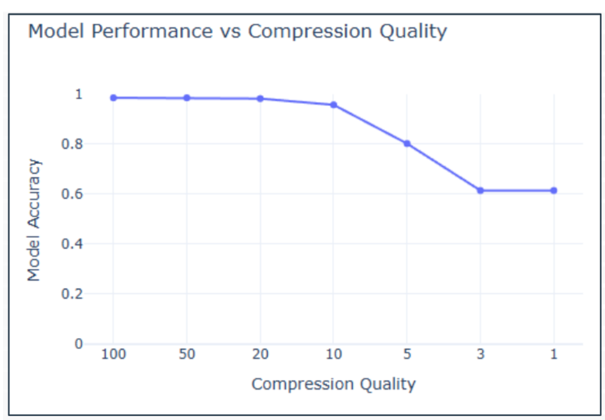

Fig. 4. Model Accuracy vs Compression Rate

The model's performance stays constant even for significant compression ratios (50% - 20% quality), implying that visual features learned from high quality images transfer successfully to same-distribution images with up to 80% less information. Interestingly, we see a sharp decrease in accuracy past 10% quality, which matches the common rule-of-thumb that 1:10 data compression is reasonable threshold for compression without noticeable artifacting [1].

IV. Conclusions

Our study shows that a model's learned visual features successfully transfer to compressed images when classifying various skin conditions. The quality of a model also decays in a similar manner to human evaluation of said conditions with extreme compression rates (quality $< 10\%$). Further research is needed to determine whether a model trained on compressed images learns features which generalize to uncompressed images. It would also be of interest to conduct a thorough analysis of a network's saliency maps to determine at which point a model's internal representations are broken down by compression artifacts.

References

[1] Richard F. Haines and Sherry L. Chuang. "The effects of video compression on acceptability of images for monitoring life sciences experiments". In: *IEEE Computer Society Data Compression Conference* (1992).

[2] M et al. Hersche. "Constrained Few-shot Class-incremental Learning". In: *CVPR 2022* (1973).

[3] Graham Hudson et al. "JPEG-1 standard 25 years: past, present, and future reasons for a success". In: *Journal of Electronic Imaging* (2018).

[4] OT Jones. "Recognising Skin Cancer in Primary Care". In: (2020).

[5] P Kanerva. "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors". In: *Cognitive Computation* (2009).

[6] Zewen Li et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.

[7] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[8] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.