**Transfer Learning with Popular Image Classification Models to Diagnose Alzheimer's Disease**

Dawson Haddox, Ben Williams, Michael Maddison, Will Balkan, Brian Chun Yin Ng

May 14th, 2024

**1. Abstract**

We explored the efficacy of transfer learning with popular CNN architectures for classifying Alzheimer's from MRI images in the OASIS Alzheimer’s Detection dataset. We tested the following models: Resnet, Densenet, Squeezenet, Mobilenet, and Vision Transformer. We only used 40% of the entire OASIS dataset when creating training/test/validation splits due to compute and time limitations. During experimentation, slight accuracy improvements were observed from data augmentation techniques like cropping, resizing, and hippocampus segmentation. No improvements were seen after oversampling and data trimming/balancing. Overall, the models were largely biased toward predicting Non Demented, even when using oversampling and data trimming/balancing. Vision Transformer performed the best after fine-tuning, achieving an accuracy of 80.80% in Alzhiemer’s multi-class classification.

**2. Introduction**

Alzheimer’s disease (AD) is the most common neurodegenerative and dementing disease [1]. The detection, especially the early detection, of AD is crucial in that it can delay the onset of dementia, increase the chances of benefitting from treatment, and lower cost to healthcare systems [2]. MRI measures are commonly used in AD diagnosis and classification for being non-invasive, less expensive, and widely available in most medical environments [1]. As such, the detection and classification of AD from neuroimaging data through machine learning and deep learning techniques has been a subject of intense research in recent years [3].

**3. Preliminaries**

Here we will discuss the MRI brain imagery dataset we used, as well as briefly summarize the specifications of each model that we modified. In general, we chose models that we hoped would be more adaptable to a new dataset due to how their architectures were designed.

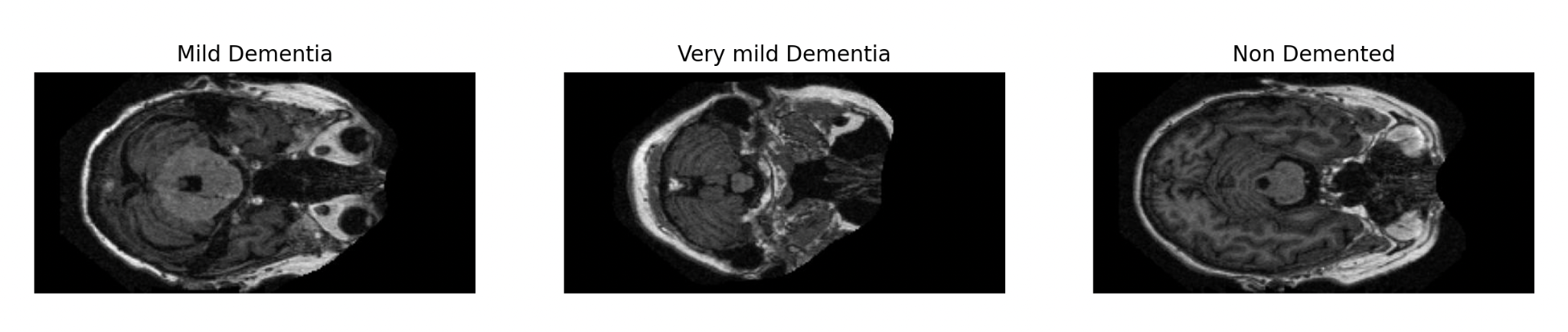
**3.1 The Dataset**

The OASIS Alzheimer’s Detection dataset is made up of over 86,000 brain MRI images. These images are divided into four categories based on the Alzheimer’s progression. The categories are moderate dementia, mild dementia, very mild dementia, and non-demented. The classes are skewed such that the vast majority of images fall into the non-demented category, and only a smaller subset of images are in the demented categories:

| **Class Label** | Moderate dementia | Mild dementia | Very mild dementia | Non-demented |
| --- | --- | --- | --- | --- |
| **Number of Images** | 488 | 5,002 | 13,725 | 67,222 |
| **% of total images** | 0.56% | 5.79% | 15.88% | 77.77% |

*Figure 1: The number of images in each class of dementia, representing the class imbalance of the dataset*

These images come from 461 different patients, so a single patient will have many brain images in each one of the categories. Due to the sparsity of the images in the moderate dementia category, the class was discarded in favor of classifying on the other three. Below is an example of the images in the data:



*Figure 2: Example brain MRI images from each category used in this experiment*

**3.2 MobileNetV3 Models**

The third version of the small and large mobilenet models were used for this experiment [4]. Both were built to be relatively lightweight image classifiers, and were trained on ImageNet. Both leverage many stacked bottleneck blocks, which leverage both linear and nonlinear transformations as well as depthwise separable convolutions. Depthwise separable convolutions are a more efficient alternative to a typical convolutional layer; they perform almost as well as regular convolutions but cost much less. This allowed the MobileNet developers to include more layers while still having a cheaper cost. The bottleneck layers each expand the input into a larger dimension, before shrinking it back down into a smaller one. This allows them to more slowly expand the dimensionality over the course of the model, while still gaining a lot of information in each layer. They also utilize squeeze-and-excite layers. More details behind these bottleneck layers can be found in the MobileNetV2 paper where they were first used [5].

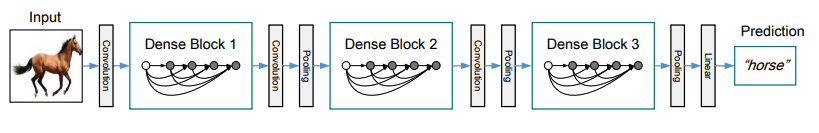
We believed that the MobileNet models had potential on our dataset as they were not excessively deep. Because of this, the pretrained models would hopefully be less overfit to the specific dataset, and we would be making more significant modifications proportionally to this model by only modifying the final fully connected layers.

**3.3 ResNet**

ResNet was designed to address degradation and vanishing/exploding gradients in very deep neural networks [6]. To do this, they added residual blocks to a traditional convolutional neural network in order to preserve information between earlier layers and future layers. This is done by having an identity mapping of one layer that shortcuts a few layers down, preventing information from being lost due to too many transformations. This allowed the ResNet developers to create a much deeper neural network while avoiding degradation. We believed that this would work well for our new dataset, as some un-transformed information about our images will make it down to the final layers. This would allow the model’s final layers to learn more information that was specific to our dataset, and not just ImageNet (which ResNet was trained on), than other standard convolutional neural networks. The model used for our experiment was the ResNet18 model, which is relatively shallow, so it aligned with our computational resources and was hopefully less overfit to the ImageNet dataset.

**3.4 DenseNet**

The DenseNet architecture leverages several dense blocks, where each dense block is made of many convolutional layers [7]. However, the information is passed between the layers inside each dense block, allowing information from earlier layers to be re-accounted for again in later layers. By creating these short paths from earlier layers to later layers, the developers aimed to address the problem of information and gradients vanishing after passing through many layers.

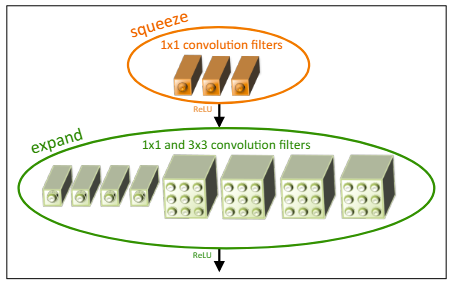


*Figure 3: An illustration of a DenseNet model with three dense blocks. The convolution and pooling between dense blocks are referred to as transition layers in their paper. [7]*

The model used for our image categorization was the DenseNet121, which consists of four dense blocks, each having between 6 to 24 dense layers, with each dense layer having two convolutional layers. DenseNet121 was trained on ImageNet. We believed that this model could perform well on our dataset because the structure of the dense blocks ensures that less data is lost throughout the model, meaning that basic features of the MRI imagery would hopefully be accounted for in the final layers, reducing the overfitting effect that the pretrained weights might have on the other dataset.

**3.5 Squeezenet**

The design of SqueezeNet derives from 3 design strategies which aim to reduce the quantity of model parameters while maintaining accuracy. These are: (1) Replace 3x3 filters with 1x1 filters, (2) Decrease the number of input channels to 3x3 filters, and (3) Downsample late in the network so that convolution layers have large activation maps [8].



*Figure 4: Microarchitectural view: Organization of convolution filters in the Fire module. [8]*

SqueezeNet leverages the Fire module to implement strategies (1) and (2). The Fire module consists of a squeeze layer including only 1x1 convolutional filters, followed by an expand layer featuring both 1x1 and 3x3 convolutional layers. To implement strategy (3), max-pooling with a stride of 2 was used at relatively later layers in the model architecture.

SqueezeNet was selected because its lower computational requirements (50x fewer parameters than AlexNet with comparable performance) are ideal for our limited computational resources.

**3.6 Vision Transformer**

The Vision Transformer model leverages the transformer architecture used for natural language processing with minimal modifications [9]. It struggled when trained on smaller datasets, but performed extremely well when used on large datasets. The model used for our research was the base version with a 16x16 patch size. It was trained on the large ImageNet dataset, and has over 86 million parameters. We believed that this model’s structure differed from the others so significantly that it would be worth experimenting with. Furthermore, the transformer architecture may be more able to focus on the parts of the brain that are the most important.

**4. Proposed Method**

**4.1 Fine-Tuning**

Each model in section (3) was fine-tuned to the MRI dataset by fixing all parameters in the existing model, then replacing the final layer with a multi-way classifier with 3 output units. Some experiments allowed the parameters of the penultimate as well as the final layer to be learned during backpropagation. We also tried adding additional convolutional layers and non-linear activations at the end of existing models. In particular, we tried using softmax as the final layer which we hoped would improve accuracy. A random seed was set for reproducibility and ease of comparison across training modifications.

**4.2 Data Preprocessing**

Given the limited data available on the Moderate Dementia class (only two participants), it was discarded. Each model took 224 x 224 x 3 rgb images as input, while the MRI data consisted of 496 x 248 grayscale images. The MRI images were cropped to reduce empty space and limit distortion during image resizing. The single grayscale value was copied into each of the 3 rgb channels. Given our compute and memory constraints, only 40% of the entire dataset was used, with train/validation/test splits segmented by participant to prevent data leakage.

**4.3 Data Augmentation**

A variety of data augmentation techniques were employed to increase accuracy and eliminate bias from the models. The Mild Dementia and Very Mild Dementia categories only comprise a combined 21.7% of the data. In order to prevent the model from becoming biased towards one class, we used two approaches involving (a) oversampling from the minority classes and (b) trimming and balancing data from across the classes to create a dataset with evenly sized classes. Further, we segmented the images to include only the hippocampus region as input, as this region is responsible for memory and would thus be most impacted by Alzheimer’s. Finally, we trained on a subset of the data composed of images with the highest image entropy to increase diversity and highlight the differences between each class, and compared this performance to a randomly selected subset of images.

**4.4 Hyperparameter tuning**

A number of hyperparameters were tested. A hyperparameter search was conducted over the Adam as well as SGD optimizers, along with different values for number of epochs, patience, and learning rate.

**5. Experiments**

We trained each model for up to 15 epochs, fine tuning the last layer of each model. Attempts to finetune more than one layer did not result in better accuracy and often made performance worse. In particular, we were hopeful that the addition of a softmax layer at the end of the final layer would improve accuracy due to this being a multi-class problem (softmax was no longer present in other models after dropping the final layer for fine tuning), but this too didn’t yield any improvements to accuracy.

We noticed slight improvements from cropping and resizing the input images, and tested segmenting the images to include only the hippocampus region as input, as this region is responsible for memory and would thus be most impacted by Alzheimer’s. This strategy resulted in slight accuracy improvements as well.

Neither oversampling nor data trimming and balancing resulted in significant improvements on the model’s accuracy. As shown in Figure 5, the model remained biased towards the Non-Demented class even when attempting to account for class imbalance.

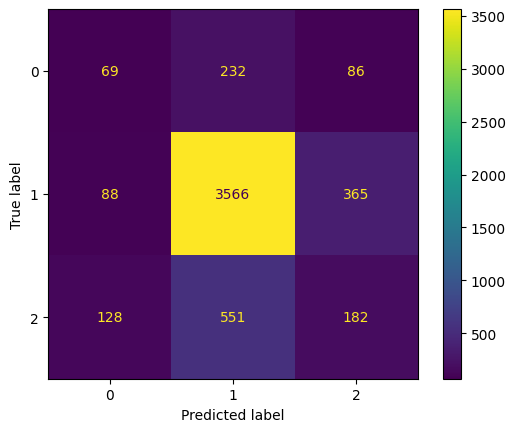
In addition to training on a random subset of images, we also tested training on a subset of images with the highest image entropy with the intention of using images that emphasize differences between the classes. However, there was no noticeable difference in model performance between the two subsets, so the random subset was used for most experiments.

As seen in Figure 5, the Vision Transformer model performed the best, followed by Densenet and ResNet18. Notably, these three models were the only to surpass the 77.77% accuracy threshold which would be obtained by weighted guessing, since 77.77% of the samples were in the Non-Demented class. Mobilenet V3 Small outperformed Mobilenet V3 Large on this dataset, although neither achieved the 77.77% benchmark.

|  | Test Accuracy | Test F1 |
| --- | --- | --- |
| Mobilenet V3 Small | 76.55% | 0.7345 |
| Mobilenet V3 Large | 75.95% | 0.7786 |
| Densenet | 79.83% | 0.7564 |
| ResNet18 | 78.83% | 0.7509 |
| SqueezeNet | 76.31% | 0.6605 |
| Vision Transformer | 80.80% | 0.7653 |

*Figure 5: Table of best accuracies and F1 scores from each architecture.*

The models were largely biased toward predicting Non Demented, even when using oversampling and data trimming/balancing. Figure 6 shows a confusion matrix for a typical training run on MobileNet V3 Small. Even with oversampling, prediction is heavily biased towards class 1, Non Demented. This represents an issue in real-world use cases since a false negative is generally more harmful than a false positive when trying to detect Alzheimer’s.



*Figure 6: Confusion matrix for a typical training run on Mobilenet V3 Small with oversampling. Class labels are: 0: Mild Dementia, 1: Non Demented, 2: Very mild Dementia*

**6. Related Work**

The majority of research groups solve the problem of classifying AD from neuroimaging data by either (1) creating a novel CNN architecture [10] or (2) duplicating an existing CNN architecture and training it from scratch [11, 12]. Such techniques have performed well on neuroimaging data, with both groups reaching Alzheimer’s multi-classification testing accuracies over 90%. However, common limitations with such algorithms that train from scratch are twofold. First, these models rely on a large amount of annotated neuroimaging data. Such data is both expensive to produce, since it relies on physicians to annotate the images, and also runs into privacy issues. Second, training CNNs with a large amount of images requires a huge amount of computational resources [3].

An alternative to training from scratch is transfer learning. Transfer learning involves taking a model pre-trained on a large dataset and fine-tuning it for a different but related task. The typical approach for fine-tuning involves freezing initial layers of the pre-trained model, since these early layers often capture general features like edges or textures, and retraining/replacing the final layers, which are often specific to the original task. In this way, transfer learning leverages the features and weights learned from the source domain, but adapts the model to the target domain.

Several groups found some success with fine-tuning popular image classifiers on MRI images through transfer learning. One group tested two models: a 2D ResNet18, a standard CNN architecture for image classification, pre-trained on the ImageNet dataset and a 3D variant of DenseNet121, an updated version of ResNet [13]. After fine-tuning the two CNNs with a low amount of data, precisely 10% and 25% of the total OASIS dataset, they achieved a peak test accuracy of 80.4% via a 3D DenseNet121. This demonstrated a reasonable amount of success for fine-tuning existing CNN architectures using low data schemes for the purpose Alzhimer’s multi-class classification and motivated our approach of fine-tuning using a low data scheme.

Though there has been some success in fine-tuning existing CNNs for Alzhiemer’s multi-class classification, there is a lack of exploration to discover which CNNs are/are not best suited for the task. Moreover, some groups use erroneous train/validation/test splits which leads to data leakage [14] and thus skewed test accuracy scores. As a result, we hope to explore fine-tuning with a larger range of CNN architectures using a low data schema.

**7. Conclusions**

We explored the efficacy of transfer learning with popular CNN architectures for classifying Alzheimer's from the OASIS Alzheimer’s Detection dataset. Our models struggled to surpass the accuracy from just weighted guessing, and when they did, such as in the case of the Vision Transformer, it was only by a small margin. We found that cropping and resizing the input images without oversampling was the most effective training method.

Our work had several limitations. Our model choices were heavily influenced by our limited computational resources. Deeper, more complex models might have performed better, but this remains uncertain.

For future work, we recommend training a model from scratch on the full dataset, rather than just fine-tuning the final layers of a pre-trained model. While more computationally expensive, this approach would likely yield much better performance on this challenging task.

**8. Division of Work**

Dawson Haddox:

* Started the project by downloading the data from Kaggle, writing programs to subset to a random forty-percent and forty-percent based on image entropy, and writing a program to create test, train, and validation splits by participant.
* Reviewed relevant literature to decide on the specific project idea and come up with potential strategies.
* Wrote the code for the ResNet model pipeline, which was largely copied for the other models
* Programmed and experimented with a variety of hyperparameters, transformations, and data trimming and balancing
* Ran experiments and model pipelines written by other group members and helped with hippocampus segmentation
* Contributed heavily to the PowerPoint and helped present it due to his familiarity with the code, experiments, and use case.

Ben Williams:

* Wrote code for the DenseNet and MobileNet models
* Wrote code to optimize running many of the models we were experimenting with so that we could run them locally with better GPUs
* Reviewed, researched, and wrote about the models that we used for the experiments
* Ran many of the experiments locally, fine-tuning and experimenting with hyper-parameters, data augmentation methods, and model sizes. One example of this was testing out fine tuning more than just the final fully connected layer of both the MobileNet Small and MobileNet Large models.
* Contributed significantly to Powerpoint and was one of the two that presented it due to familiarity with the code, models, and results

Michael Maddison:

* Reviewed Alzhimer classification literature to decide on a novel project idea. Conducted thorough research on previous strategies of classifying Alzhiemer through novel models, existing CNN architectures that are trained from scratch on MRI images, and fine-tuning existing CNN models. Then helped to find our area of novelty which in part was to test transfer-learning on a broader range of CNNs.
* Ran several experiments locally. Mainly researched and experimented with data augmentation methods, particularly hippocampus segmentation as a hope to improve the model’s test accuracy.
* Wrote the related works section on the powerpoint
* Wrote the abstract, introduction, and related works, on the paper
* Went to office hours to get help to further improve performance on our models

Will Balkan:

* Researched and found dataset for Alzheimer’s MRI images
* Wrote code for Densenet model and oversampling
* Experimented locally with different hyperparameters, especially with Densenet and Mobilenet models
* Wrote experiments, method, and parts of model descriptions after researching and reviewing models
* Compiled references

Brian Chun Yin Ng:

* Wrote code for both the Vision Transformer and Squeezenet models
* Specifically read papers on what can be done to unfreeze and finetune the final layers of existing models and adding additional layers to add model complexity and accuracy
* Programmed and experimented with a variety of hyperparameters, data trimming and balancing and tried adding additional nonlinear layers to the model without adding much computational load
* Tried to maximize model performances with Google Collab Pro+ and optimize code to run on GPUs but running on local device turned out to be significantly faster for our models
* Wrote the motivation, novelty and problem definition on the powerpoint
* Wrote the conclusion and parts of the experiments section on the paper
* Went to office hours to get help to further improve performance

**9. References**

[1] AbdulAzeem, Y., Bahgat, W.M. & Badawy, M. A CNN based framework for classification of Alzheimer’s disease. Neural Comput & Applic 33, 10415–10428 (2021). https://doi.org/10.1007/s00521-021-05799-w

[2] Rasmussen, Jill, and Haya Langerman. “Alzheimer's Disease - Why We Need Early Diagnosis.” Degenerative neurological and neuromuscular disease vol. 9 123-130. 24 Dec. 2019, doi:10.2147/DNND.S228939

[3] M. Hon and N. M. Khan, "Towards Alzheimer's disease classification through transfer learning," 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, 2017, pp. 1166-1169, doi: 10.1109/BIBM.2017.8217822.

[4] Howard, Andrew, et al. "Searching for mobilenetv3." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[5] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[6] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[7] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[8] Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." *arXiv preprint arXiv:1602.07360* (2016).

[9] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[10] S. Basheer, S. Bhatia and S. B. Sakri, "Computational Modeling of Dementia Prediction Using Deep Neural Network: Analysis on OASIS Dataset," in IEEE Access, vol. 9, pp. 42449-42462, 2021, doi: 10.1109/ACCESS.2021.3066213.

[11] Yadav, K. S. & Miyapuram, K. P. A novel approach towards early detection of alzheimer’s disease using deep learning on magnetic resonance images. In Brain Informatics: 14th International Conference, BI 2021, Virtual Event, September 17–19, 2021, Proceedings 14 (Springer, 2021).

[12] Fuadah, Y. N. et al. Automated classification of alzheimer’s disease based on MRI image processing using convolutional neural network (CNN) with AlexNet architecture. J. Phys. Conf. Ser. 1844(1), 012020. https://doi.org/10.1088/1742-6596/1844/1/012020 (2021).

[13] Nikhil J. Dhinagar, Sophia I. Thomopoulos, Priya Rajagopalan, Dimitris Stripelis, Jose Luis Ambite, Greg Ver Steeg, Paul M. Thompson, "Evaluation of transfer learning methods for detecting Alzheimer’s disease with brain MRI," Proc. SPIE 12567, 18th International Symposium on Medical Information Processing and Analysis, 125671L (6 March 2023); https://doi.org/10.1117/12.2670457

[14] Aithal, Ninad. (2023; July). OASIS Alzheimer's Detection, Version 1. Retrieved May, 2023 from https://www.kaggle.com/datasets/ninadaithal/imagesoasis/data.