[Submission Guidelines](https://www.springer.com/journal/521/submission-guidelines?gclid=CjwKCAjwjZmTBhB4EiwAynRmD_IEjJN6ZvvnyUJu16--mSlv7BM34-MiXnCGRRToR-P6QrhBMVnNIRoCkEQQAvD_BwE)

Recommended order for writing an original manuscript

1. Tables and Figures – each have a clear point, tell story of manuscript
2. Results – falls out of tables and figures
3. Methods – play by play of what you did
4. Introduction – tables help you frame introduction
5. Discussion -- hardest section
6. Abstract – do this last

# Abstract:

Lack of data is an on-going problem in Alzheimer’s disease (AD) research. AD can be hard to diagnose and even experienced neurologists struggle to read MRIs accurately. While many studies have used CNN models to classify MRI images, this study compares flipping and shifting images to determine which type of data augmentation is most helpful. This study also compares three different pre-trained CNN models (ResNet50, VGG16, and Xception) as well as an ensembled model. All three models easily obtained excellent (99%) accuracy on the training/validation sets, even without augmentation, thought test set accuracy lagged behind (65-73%). Shifting images improved test set accuracy by 2-7%, depending on the model. Ensembling the three models together improved test set accuracy by another 8-11%, depending on the model. Augmenting images to increase and balance datasets, as well as ensembling well performing models, hold great promise in improving machine learning models to diagnose AD.

# Introduction

Alzheimer’s disease is the most common dementia, yet it remains difficult to diagnose. PET scans to measure amyloid protein build-up and cerebrospinal fluid (CSF) biomarkers can help to provide an accurate diagnosis, however they are expensive and invasive tests. MRIs are much more accessible.

However, MRIs are difficult to read and depend on an experienced neurologist to interpret them. A dependable and accurate ML learning model has the potential to help a general practitioner in a small town to read an MRI as accurately as a specialist at a large research hospital. There has been progress in developing ML models for other conditions like breast cancer. [2](https://paperpile.com/c/wOWd5O/lxpf) [Yala]

Being able to differentiate mild cognitive impairment from normal cognition and Alzheimer’s disease is also important for clinical trials, as potential treatments are more likely to be effective before the condition progresses to AD.

In the past, neurological ML models mainly used MRIs for feature extraction, i.e. to get a measurement for a part of the brain, such as the size of the hippocampus. While this has proved helpful, there is significant domain knowledge required in order to understand which parts of the brain to target. Extracting features is challenging; we also can’t be sure that we are picking out the most helpful features.

CNNs offer a different approach of using computer vision to look at the image as a whole. There is less domain knowledge needed as the computer decides which are the relevant features.

However, using a CNN network is not without its own challenges. CNN networks work best when you have lots of data; however, in dementia research there is a lack of labeled images. While I’ve read about some interesting approaches, including using unlabeled data to help train a network [3](https://paperpile.com/c/wOWd5O/zRul), the majority of researchers continue to use labeled datasets.

One way around a lack of data is data augmentation. In this paper, three different approaches are compared: using the original non-augmented dataset, using a flipped training set, and using a shifted training set.

Another method of dealing with a lack of data when creating a CNN model is to use transfer learning, i.e. using pretrained networks as a starting point for your model.While there has been some discussion about whether it is useful to use networks pre-trained on ImageNet with medical data [4](https://paperpile.com/c/wOWd5O/B5Hy), many studies that have used these pretrained models to look at Alzheimer’s imaging data. [5](https://paperpile.com/c/wOWd5O/pM6X) [6](https://paperpile.com/c/wOWd5O/Pzkv) [7](https://paperpile.com/c/wOWd5O/1uMv)

In this study, three common models (LIST) will be looked as, and well as combined, to try to quantify the gain that can be had by ensembing models.

# Methods

A Kaggle dataset was used for this project: [Alzheimer’s Dataset (4 class of images).](https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images)  This dataset contains approximately 5000 jpg images (slices) taken from MRI images of possible Alzheimer’s patients, divided into 4 categories: non-demented, very mild demented, mild demented, and moderate demented.

The original dataset was quite unbalanced, although typical for this type of research. As patients become more impaired, there is often less data available.

|  |  |
| --- | --- |
| Train Set | # Images |
| Non-impaired (cognitively normal) | 2560 |
| Very mild impairment | 1792 |
| Mild impairment | 717 |
| Moderate impairment | 52 |

|  |  |
| --- | --- |
| Test Set | # Images |
| Non-impaired (cognitively normal) | 640 |
| Very mild impairment | 448 |
| Mild impairment | 179 |
| Moderate impairment | 12 |

The train set was divided into a train/validation, using a 70/30 split. Thirty percent of the training images were moved to a separate validation folder. The flipping and shifting transformations were only applied to the train set. The final non-augmented dataset consisted of:

|  |  |
| --- | --- |
| Train Set | # Images |
| Non-impaired (cognitively normal) | 1792 |
| Very mild impairment | 1254 |
| Mild impairment | 502 |
| Moderate impairment | 36 |

|  |  |
| --- | --- |
| Validation Set | # Images |
| Non-impaired (cognitively normal) | 768 |
| Very mild impairment | 538 |
| Mild impairment | 215 |
| Moderate impairment | 16 |

|  |  |
| --- | --- |
| Test Set | # Images |
| Non-impaired (cognitively normal) | 640 |
| Very mild impairment | 448 |
| Mild impairment | 179 |
| Moderate impairment | 12 |

**Flipping the Data**

The original jpg images were flipped using cv2.flip() in three directions: horizontally (using flip code 1), vertically (using flip code 0) and around both axes (using flip code -1). The data was also balanced for all three classes, except for the smallest minority class, by selecting varying numbers of the original images to flip. The smallest class did not contain enough images for it to be balanced.

|  |  |
| --- | --- |
| Flipped Train Set | # Images |
| Non-impaired (cognitively normal) | 2008 |
| Very mild impairment | 2008 |
| Mild impairment | 2008 |
| Moderate impairment | 144 |

The validation and test sets for the flipped data remained the same as for the non-augmented data.

**Shifting the Data**

The original jpg images were also shifted, using cv2.warpAffine(). By using more shifts on the smallest class, a balanced dataset was created. This seemed like it might be a good transformation to try because MRIs are all taken at slightly different angles, depending on the machine and the person.

|  |  |
| --- | --- |
| Shifted Train Set | # Images |
| Non-impaired (cognitively normal) | 2008 |
| Very mild impairment | 2008 |
| Mild impairment | 2008 |
| Moderate impairment | 2016 |

After augmenting the data, we had 3 different training sets: one with no augmentation, one with flipping the images, and one with shifted images. The validation and test sets remained the same.

**Pre-trained CNN Networks**

Three pre-trained CNN networks were selected for this project: ResNet50, VGG16, and Xception. ADD LINKS TO THESE MODELS

ResNet50 was the first CNN model to introduce the concept of a skip connection: a signal feeding into a layer is also added to the output of another layer that is located higher up in the stack. It won the ImageNet challenge in 2015. Input shape (224, 224, 3)

VGG16 won the ImageNet competition in 2014 and uses smaller filter, but a deeper network. It was named after the Visual Geometry Group from Oxford that developed it. Input shape (224, 224, 3)

Xception used depthwise separable convolution and improves upon the Inception model. It dates from 2017 and is 71 layers deep. Input shape (299, 299, 3)

**Creating the Models**

Each pre-trained model was run on each dataset, for a total of nine models. Finally, the best three models were selected and ensembled together using a standard random forest algorithm. See Figure 1 for an example of how this was done.

Diagram

Description automatically generated

# Results

## Individual Model Results

All the models were trained on Google Colab Pro, using a batch size of 32, with dropout of 0.1. The frozen layers were trained for 10 epochs (20 epochs for Shifted VGG16 model) with a learning rate of 0.001, using both early stopping and reduce LR on plateau callbacks. Each model was then unfrozen and trained it for an additional 10 epochs, starting with a learning rate of 0.0001. See Table 1 for results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Augmentation | Train Acc | Val Acc | Test Acc |
| ResNet | None | 1.00 | 0.9740 | 0.7303 |
| VGG16 | None | 0.8597 | 0.8699 | 0.6529 |
| Xception | None | 0.9980 | 0.9453 | 0.6701 |
|  |  |  |  |  |
| ResNet | Flip | 0.9992 | 0.9818 | 0.7224 |
| VGG16 | Flip | 0.9162 | 0.8530 | 0.6482 |
| Xception | Flip | 0.9990 | 0.9655 | 0.7177 |
|  |  |  |  |  |
| ResNet | Shift | 0.9981 | 0.9876 | 0.7482 |
| VGG16\* | Shift | 0.9999 | 0.9759 | 0.7240 |
| Xception | Shift | 0.9994 | 0.9681 | 0.7209 |
|  |  |  |  |  |
| Combined |  |  |  | 0.8307 |

\*trained 20 epochs, as I realized this model was taking longer to train

## Ensembled Results

To ensembled the models, I took the individual CNN model with the best results – this was the shifted models worked for all three of the pre-trained models. I created a data frame of the results (see sample below), and then trained it again using a random forest, with a 70/30 train test split.

Graphical user interface, text, application

Description automatically generated

Here are the final results of that model.

A picture containing text, receipt, screenshot

Description automatically generated

The confusion matrix below provides even more info.

Graphical user interface, application

Description automatically generated

# Discussion

Each of the pretrained models did an excellent job on the training and validations sets; all of them struggled to deal with the training set.

It is interesting to note that flipping the images did NOT help the models; in fact, for two of the models, ResNet and VGG16, it made them worse. This is probably due to the fact that all these images were already standardized to face in one direction. It is also unclear if you can flip images of the brain – different hemispheres play different roles. It is possible that flipping the images actually confuses the models, because it now sees changes in a different part of the brain.

Shifting our images, however, was helpful. It improved the test accuracy for all three of our models – improving our ResNet model by about 2%, the Xception model by 5%, and the VGG16 model by 7%. See table \_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Test Acc (No Aug) | Test Acc  (Shifted) | Improvement |
| ResNet | 0.7303 | 0.7482 | 0.0179 |
| VGG16 | 0.6529 | 0.7240 | 0.0711 |
| Xception | 0.6701 | 0.7209 | 0.0508 |

The models also varied in how well they classified each category.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Precision | | | Recall | | | F1-score | | |
|  | R | V | X | R | V | X | R | V | X |
| Mild | 0.82 | 0.69 | 0.80 | 0.37 | 0.40 | 0.50 | 0.51 | 0.51 | 0.62 |
| Mod | 1.00 | 1.00 | 0.56 | 1.00 | 0.17 | 0.83 | 1.00 | 0.29 | 0.67 |
| Non | 0.77 | 0.74 | 0.84 | 0.86 | 0.89 | 0.69 | 0.81 | 0.81 | 0.76 |
| Very | 0.70 | 0.70 | 0.61 | 0.73 | 0.63 | 0.84 | 0.72 | 0.66 | 0.71 |

For the “mild” class:

* ResNet had the best precision (0.82)
* Xception had the best recall (0.50) and f1 score (0.62)

For example, for the “non” class:

* Xception had the best precision (0.84);
* VGG16 had the best recall at (0.89);
* VGG16 and ResNet had similar f1 scores (0.81)

ADD Discussion of other classes

Ensembling the models results in a dramatic improvement in our accuracy for the test set data, ranging from 8-11%.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Test Acc  (Shifted) | Test Acc  (Ensembled) | Improvement |
| ResNet | 0.7482 | 0.8307 | 0.0825 |
| VGG16 | 0.7240 | 0.8307 | 0.1067 |
| Xception | 0.7209 | 0.8307 | 0.1098 |

# Conclusion

When comparing non-augmented, flipped, and shifted datasets, shifting the images to create more data improved test set accuracy – even as it had no effect on the accuracy of train and validation sets. In this study, it improved test performance by 2-7%, depending on the model used.

In a field such as Alzheimer’s research, shifting images to create more data is a way to improve test-set performance and perhaps increase the transferability of the model to other datasets.

Even when various models provide good results individually on test and validation sets, performance can be increased on the test set by ensembling them together to take advantages of their individuals strengths and minimize their weaknesses. In this study, ensembling three models improved test performance by 8-11%, even when all of the models had excellent accuracy on the training and validation sets.

Ensembling is another method that could be put to good use when dealing with small, limited datasets in neurodegenerative disease research.

Further research directions might include more detained investigations into the optional amount to shift images; exploring more pre-trained models; and/or investigating other image transformations such as brightness or normalization techniques.