Application of CNN Models in Identification of Alzheimer Syndrome Based on MRI Segmentation Images

Marina Nikolaeva, Vladimir Bliznyukov

Abstract

Alzheimer syndrome is an increasing problem of today's population. This disease causes a significant brain damage and subsequently deprives individuals of performing daily tasks. In our project we would like to use MRI segmentation scans to explore the ways of identifying the extent of disease severity with the use of CNN models. This is a classification task.

1 Introduction

Alzheimer syndrome is a disease that causes brain and memory damage and gradual disability of performing daily activities. Nowadays, Alzheimer syndrome is becoming a common disease even within younger groups. Unfortunately, today there is no official cure for this syndrome. Thus, it is crucial to develop an automated way to detect this syndrome at early stages, which will allow for coherent treatment. Moreover, automating such diagnosis procedure will remove personal judgement making it more objective and give more people access to this system from anywhere in the world.

In this project, we explore CNN models that help identify Alzheimer syndrome based on brain scans. This will help to automate diagnosis process and pave the way to objective and accessible feedback for everyone.

2 Proposed solution

The dataset will be taken from Kaggle official page link. It consists of 4 groups of participants: Mild, Moderate, VeryMild and Healthy respectively. The first objective is to effectively preprocess (dilation, erosion, edge detection, filtering) the images for reasonable feature extraction.

The second objective is to specify the practical procedure of dealing with highly imbalanced datasets (patients are the minority group). This can be achieved via upsampling augmentation techniques, whose performance we would like to compare and test (Synthetic Minority Oversampling Technique or geometric augmentation with OpenCV, Albumentations). The effectiveness of augmentation diversity can be evaluated with SSIM index.

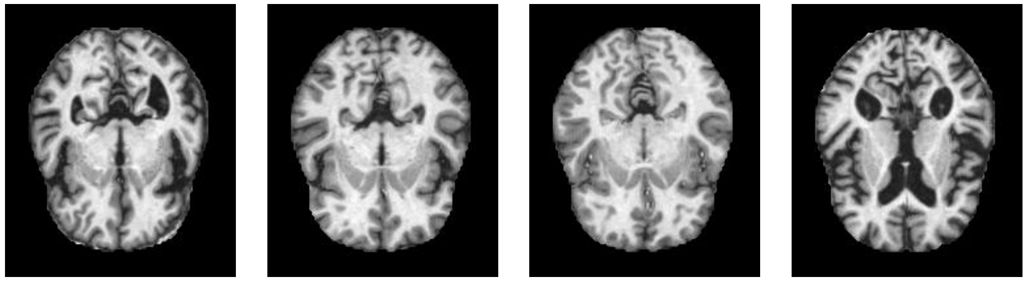
Lastly, we take an opportunity to compare performance of simple CNNs and VGG, ResNet, EfficientNet, DenseNet on the aforementioned classification task. This section would include building our own non-complex CNN (less than 20 layers) and working with tuning model layers (Batch, Dropout, Pooling) and hyper-parameters such as learning rate (or adaptive learning rate), early stopping, optimizers. Since it is a classification task, model performance would be evaluated on smoothed labels via Accuracy and AUC (TPR vs FPR) which is one of the most common metric in medical research.

By the end of this project we would expect to present the best working model with the benchmark on other configurations.

3 Alzheimer's Dataset

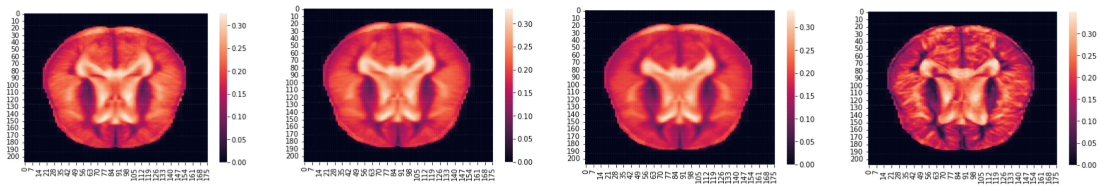
The data consists of MRI images. There are two files in dataset - Training and Testing both containing a total of around 5000 images each segregated into the severity of Alzheimer's. There are four classes:

1. Mild Demented
2. Very Mild Demented
3. Non Demented



(a) Mild Demented (b) Very Mild Demented (c) Non Demented (d) Moderate Demeneted

Figure 1: Alzheimer's disease severity classes

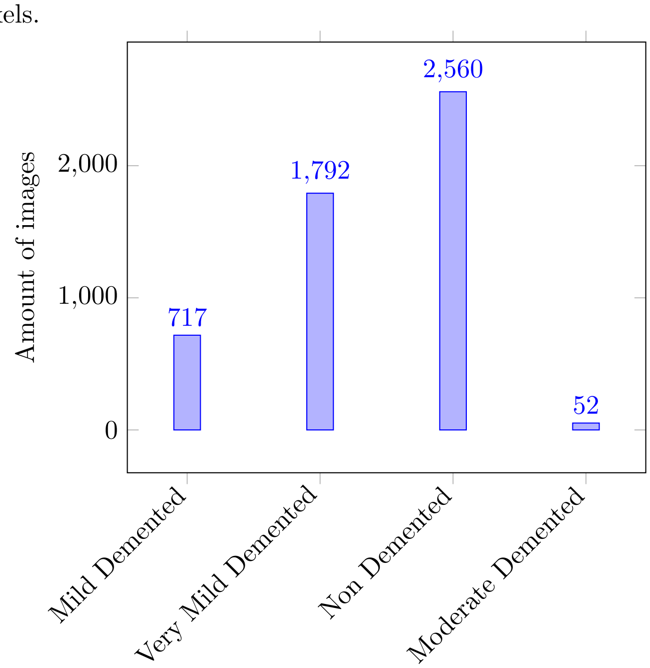


(a) Mild Demented (b) Very Mild Demented (c) Non Demented (d) Moderate Demeneted

Figure 2: Alzheimer's disease severity classes using heatmap

4. Moderate Demeneted

Mild Demented training set consists of 717 images, Very Mild Demented - 1792 images, Non Demented - 2560 images and Moderate Demented - 52 images. Each image has a resolution of 176×208 pixels.

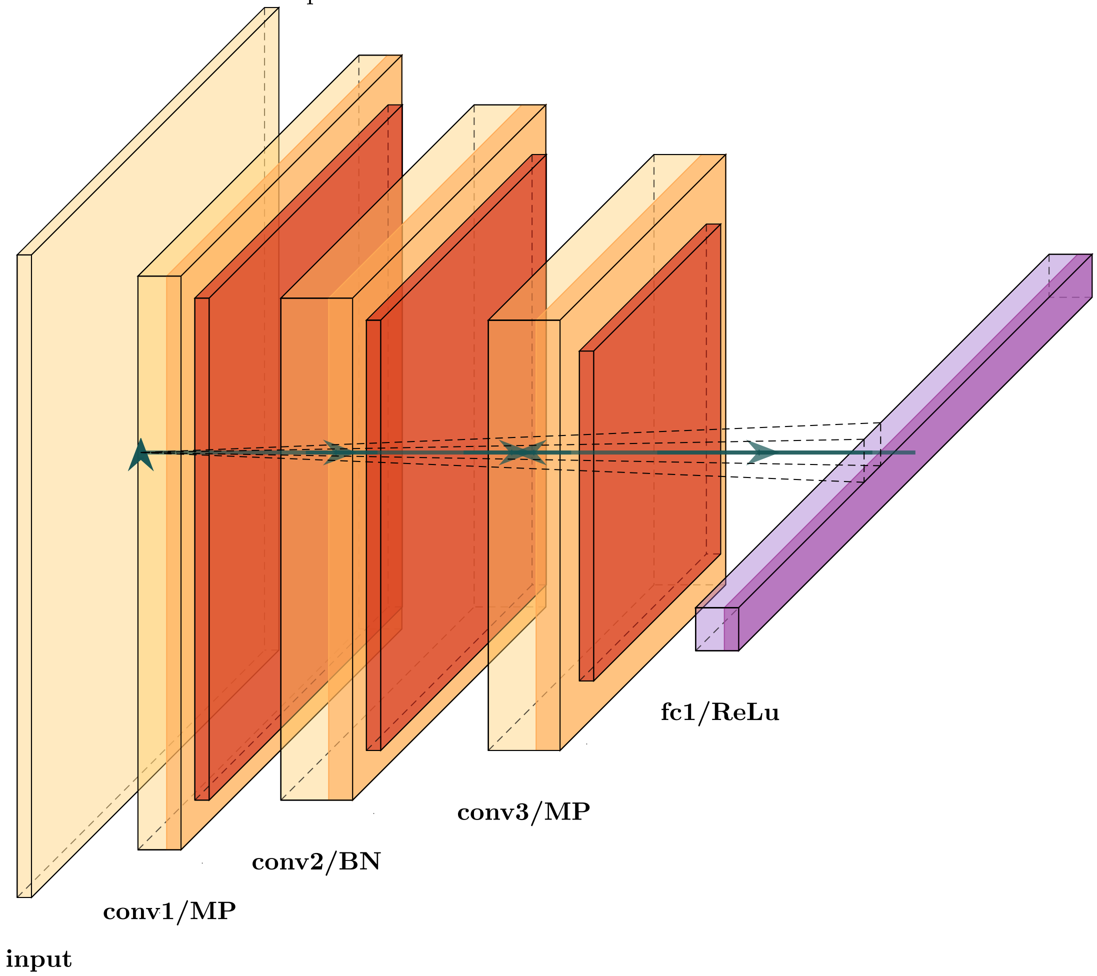


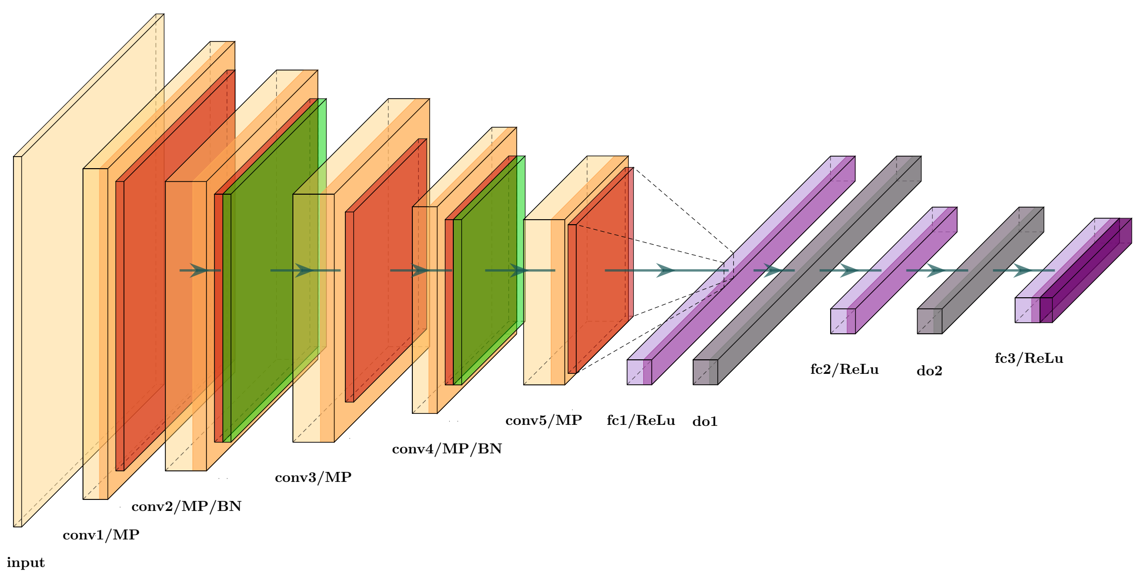
3.1 Preprocessing

As initial step for preprocessing we chose PyTorch transform function with 3 steps: Resize (for faster processing), ToTensor, Normalize.After that we loaded data from train and test (pre-splitted by Kaggle). As

4 Models

For our first iteration we wanted to look into models performance on the dataset. We chose pretrained state of art models: VGG, DenseNet, ResNet and created our own simple CNNs, namely, 3 and 5 layer CNNs. The structure of last two are presented below:





We run the models on the dataset for 15 epochs and evaluated after every epoch on the test set. Overall, each model was run 3 times for each of the optimizers. For the metric we use accuracy and Area Under Curve, which is a typical evaluation technique for medical data (accounts for TPR and FPR).

4.1 Results

Here is a preliminary table. The images were not augmented for now, and unbalanced distribution is kept. On the confusion matrices below, you can see that the best performing plain configuration is DenseNet for now. Anyway, we need take into account further model tuning.

	cnn3	cnn5	densenet	vgg	resnet
adam	56.22% (0.577)	64.19% (0.65)	77.17% (0.761)	56.68% (0.585)	65.36% (0.665)
sgd	56.37% (0.566)	71.85% (0.701)	66.69% (0.644)	76.23% (0.746)	71.62% (0.713)
rmsprop	55.9% (0.561)	59.97% (0.599)	76.08% (0.81)	58.64% (0.579)	62.31% (0.644)

Table 1: Results for CNNs Accuracy+AUC

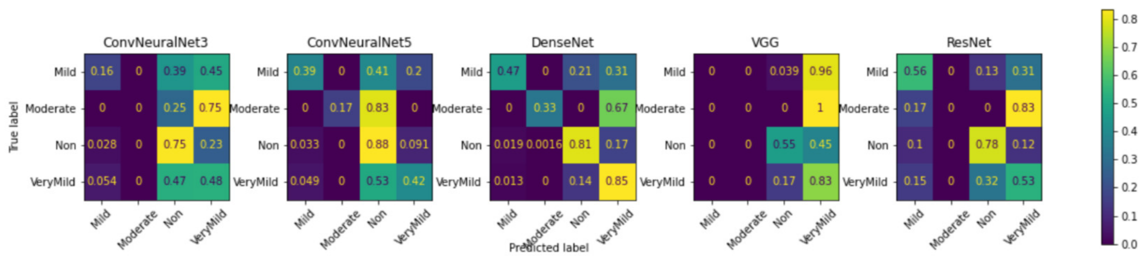


Figure 3: Confusion percentage matrices for set of models with adam optimizer

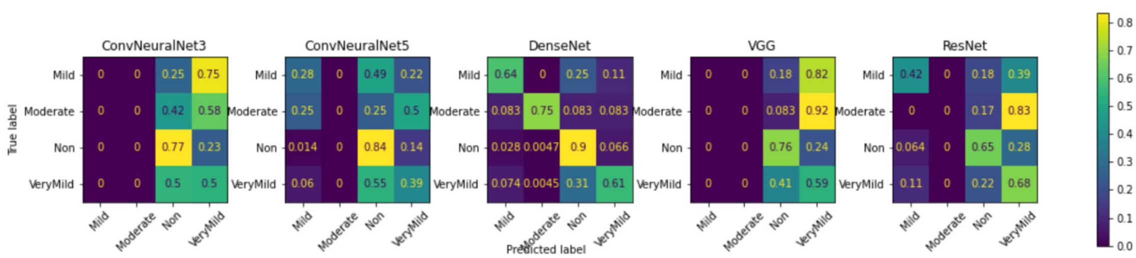


Figure 4: Confusion percentage matrices for set of models with rmsprop optimizer

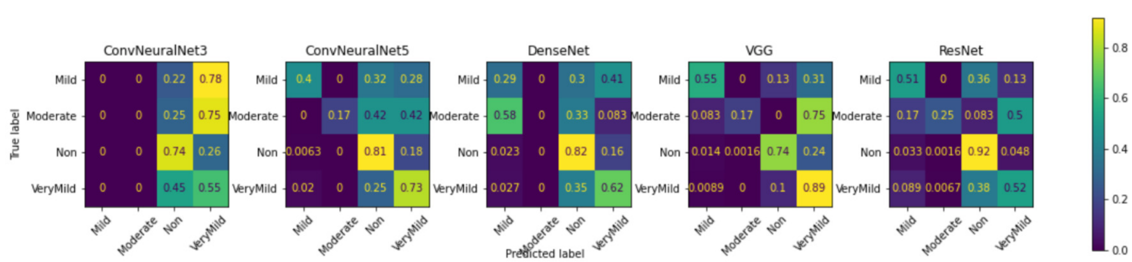


Figure 5: Confusion percentage matrices for set of models with sgd optimizer

Figure 6: Plots for RMSProp optimizer

5 Completed tasks

- Looked into the dataset
- Created simple CNNs
- Completed code for training
- Completed code for model evaluation
- Looked into pixel-wise variance of each class

6 Future tasks

- Try to apply contrast, thresholding on the target images. In this way we would highlight main image feature
- Use weighted loss for unbalanced data on best model for each optimizer
- Use more epochs, but apply early stopping, since, as one can see on the plots, some models do not improve over time.
- Try out different learning rates (Step, exponential)
- Try weighted loss on the best model (DenseNet)
- Split train into train and validation