Navigation

Alzheimer Detection

Team Members

Erin Tan Eileen Yang Wesley Tam Tong Jing Steven Li

Video Overview

finalvideo

Final Findings

Introduction & Background

Computer Aided Diagnosis (CAD) is a useful application of technology in the medical industry [3]. The use of CAD has the potential to improve and apply to many fields of medicine, especially Alzheimer's disease. If detected early, the prognosis can be greatly improved, but early diagnosis with CAD can be inconsistent given the characteristics of the disease, with accuracy rates spanning 70-95% across various models [1]. Using various classification techniques, studies have been able to detect Alzheimer's with results exceeding 90% accuracy with differentiation [2].

The MRI scans from this dataset are categorized by the level of dementia of each patient: Mild, Moderate, Very Mild, and Non-Demented Dementia.

Problem Definition

The Problem

Alzheimer's disease is a progressive neurodegenerative disorder, and early detection is critical for patient care, yet it is often diagnosed too late due to the difficulty of identifying subtle early-stage brain degeneration.

The Solution

We propose a machine learning model to analyze images for early-stage Alzheimer's detection, focusing on subtle patterns of brain degeneration that may be overlooked by doctors.

How it Differs from Prior Literature

While existing models primarily detect moderate to advanced Alzheimer's stages, our approach emphasizes early detection, which is vital for timely clinical trials and treatments.

Methods

Dataset: We will use the OASIS MRI dataset, which consists of 80,000 MRI brain scans, labeled according to the progression of Alzheimer's present in the patient. **Data Splitting**: We split all data for each model into 70% training, 20% testing, and 10% validation.

CNN Model

Preprocessing Methods:

1. **Feature Standardization**: We normalized the pixel values by dividing by 255, bringing values to a range [0,1].
2. **Brightness Adjustment**: Randomly adjusted the brightness to between 80% - 120% of the original to vary the lighting conditions.
3. **Zoom Range**: Randomly zoomed 1% either in or out.
4. **Horizontal Flip**: Flipped the photos horizontally for more variety.

Why This Algorithm?

- **Convolutional Neural Networks (CNN)**: We chose CNN because CNNs are well-suited for image classification tasks, as each layer can detect features and patterns through the use of sliding kernels, progressively recognizing larger and more complex patterns in the images.

Defining the CNN Model Layers:

1. **Rescaling**: Rescales the input data.
2. **Conv2D**: Applies convolutional layers to extract features.
3. **MaxPooling2D**: Reduces spatial dimensions.
4. **Flatten**: Flattens the input.
5. **Dense**: Fully connected layers for classification.

Logistic Regression Model

Progress During Midterm Checkpoint: During this midterm checkpoint, we worked on implementing Convolutional Neural Networks (CNN).

Preprocessing Methods:

1. **Grayscale Conversion**: Reduced complexity by having a single color channel instead of RGB.
2. **Normalization**: Converted 2D image arrays into 1D feature vectors.
3. **Standardization**: Standardized to ensure that the data had zero mean and unit variance.

4. **Undersampling**: Reduced the number of samples in majority classes to prevent the model from being biased towards majority classes.
5. **PCA**: Reduced the number of features by transforming the high-dimensional data into a lower-dimensional space while retaining variance.

Why This Algorithm?

- **Logistic Regression**: We chose Logistic Regression because it provided a baseline image classification, particularly when combined with feature extraction from the CNN model. It's faster and more efficient than SVM, making a great model for quick model iteration.
- We used L2 regularization to reduce overfitting. We used this instead of L1 regularization because we wanted to avoid feature reduction and retain them for analysis in case they are relevant. Additionally, L2 regularization is better for multicollinear data.

SVM Model

Preprocessing Methods:

1. **Grayscale Conversion**: Reduced complexity by having a single color channel instead of RGB.
2. **Normalization**: Converted 2D image arrays into 1D feature vectors.
3. **Data Type Conversion**: Converted data from float64 to float32 to half the memory consumption.
4. **Undersampling**: Reduced the number of samples in majority classes to prevent the model from being biased towards majority classes.
5. **PCA**: Reduced the number of features by transforming the high-dimensional data into a lower-dimensional space while retaining variance.

Why This Algorithm?

- **Support Vector Machine (SVM)**: SVMs are known for its performance with high-dimensional data, making it well-suited for classification tasks like determining the stage of Alzheimer's in image scans. Its ability to handle complex decision boundaries would make it a good model for challenging classification problems in the datasets. Furthermore, using class weighting as a parameter adjusted the weights inversely proportional to class frequencies, allowing for more attention to minority classes.
- We specifically chose to use LinearSVC since it processed large-scale data with high dimensions faster and offered more flexible regularization options. Using SVC was first attempted due to its suitability for both linear and non-linear classification tasks; however, it was highly inefficient on the large dataset.

Results & Discussion

CNN Results & Discussion

At the outset, we predicted that CNNs would perform the best out of the three models that we seek to implement. We implemented our CNN first to serve as a benchmark for the models that we expect to perform worse. Our CNN shows the following performance metrics when executed on a 10% testing dataset of images:

Refer to Section **Model Performance Evaluation and Visualization** in our `/notebooks/1_CNN.ipynb` file in our GitHub repository!

```
Accuracy: 0.9997686657798854
Precision: 0.9998528671972764
Recall: 0.999314672080696
F1 Score: 0.9995835054103226
```

CNN Performance Metrics

Performance Metrics:

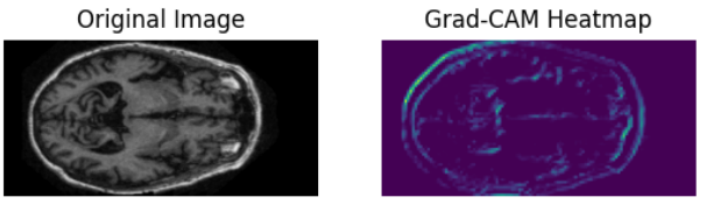
Metric	Value
Accuracy	0.999711
Precision	0.999675
Recall	0.999543
F1 Score	0.999609

We see from the metrics that the model performed extremely well on the test data, with values greater than 0.999 across all metrics. The interpretations of each metric are as follows:

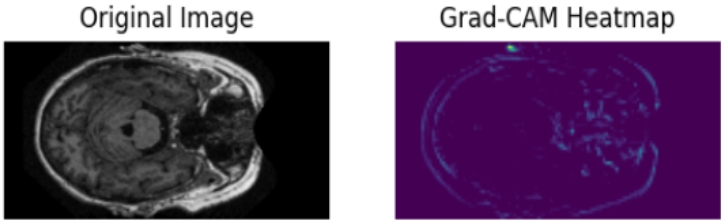
- **Accuracy**: Out of all predictions made by the model, >99.9% of predictions were correct classifications of the brain scan.
- **Precision**: Averaged across each class, out of all positive predictions made by the model for that class, >99.9% of them were correct classifications of the brain scan.
- **Recall**: Averaged across each class, out of all brain scans that were truly of that class, the model correctly classified them 99.9% of the time.
- **F1 Score**: There is a near perfect balance between precision and recall, indicating that the model is relatively free of bias.

GradCam:

After running a gradcam algorithm on the last convolutional layer of the model, we find that the model seems to focus heavily on specific locations around the brain to determine the presence of Alzheimers, as shown in the following figures:



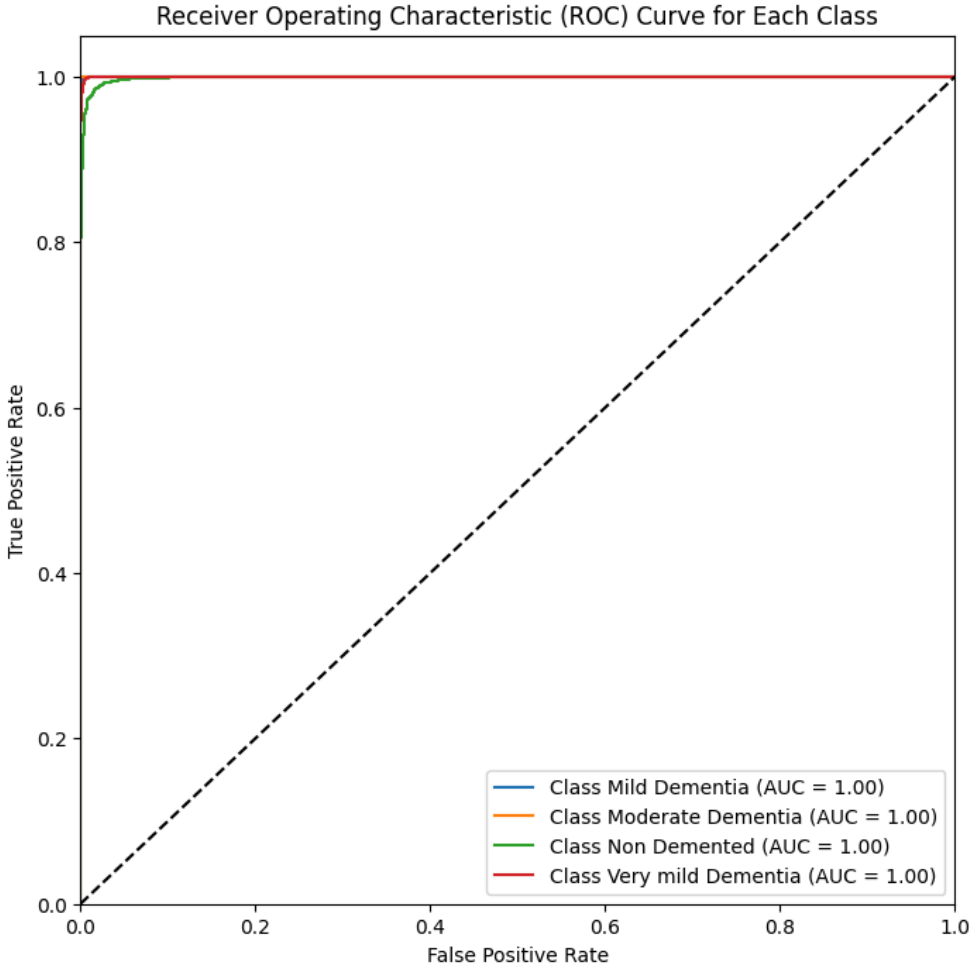
GradCam Example 1



GradCam Example 2

These figures all highlight the frontal lobe of the brain which shows that the model seems to realize that demented patients have a different brain size and frontal lobe than healthy patients. The gradcam also confirms that the model isn't looking for watermarks in the image that would label the image.

Additional Visualizations:



roc_per_class.png

Previous Next

Examining the curves, we see that when plotting true positive rate against false positive rate, we obtain a visualization of the sensitivity-specificity tradeoff in multiple ROC curves (one for each class). We note that the area under the curve for all of them is essentially 1, with the curve very closely approaching a near 100% true positive rate and a near 0% false positive rate, suggesting a highly effective model that maximizes correct positive predictions while minimizing incorrect positive predictions to a near perfect rate such that nearly all positive brain scans for a given class have a higher likelihood assignment than negative scans. This is demonstrated by the probability distributions of true positive and false positive curves, where it is evident that thresholds very close to 0 will still correctly classify nearly all positive cases. Furthermore, the learning curve and loss curve show distinctly rapid improvement, approaching the validation accuracy within the first few epochs.

The model produced unexpectedly perfect results as shown by the metrics and visualizations, raising suspicions of overfitting as it may be following training data too closely. However, these high performing metrics and results were produced when the model was run on the testing data and not the training data, which provides evidence against overfitting. We believe that this high performance is a result of all of our brain scan images coming from the same data set, which contains very similar images and very consistent labeling schemes that allows the model to succeed with the given training and testing data. As we move forward, we plan to possibly introduce a dataset with more diversity, training our model on more varied and realistic data.

Logistic Regression Results & Discussion

Logistic Regression models are good and efficient at classifying the differences between classes. We wanted to see if Logistic Regression would compare to our CNN model. We first trained a model on the original data (before balancing), then we trained a model on a balanced subset of the data(after balancing). The reasons for this are outlined in the **Comparisons and Aside on Error** section. Our Logistic Regression model shows the following performance metrics when executed on a 10% testing dataset of images:

Metric	Value before balancing	Value after Balancing
Accuracy	0.9243	0.9044
Precision	0.9217	0.9328
Recall	0.9013	0.9347
F1 Score	0.9114	0.9337

Logistic Regression Performance Metrics

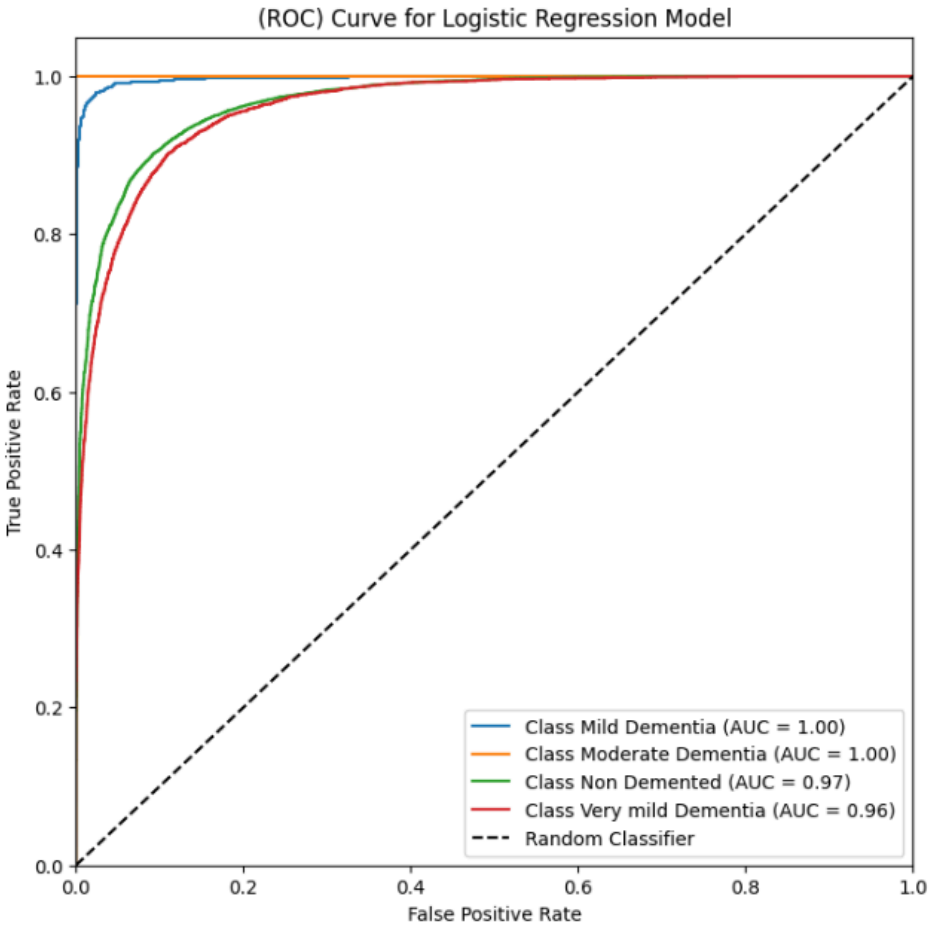
Performance Metrics:

Metric	Value Before	Value After
Accuracy	0.9243	0.9044
Precision	0.9217	0.9328
Recall	0.9013	0.9347
F1 Score	0.9114	0.9337

We see from the metrics that the model achieved a very good performance on the test data before and after we balanced the dataset, with values scoring around 0.91 across all metrics. The interpretations of each metric are as follows:

- **Accuracy**: Out of all predictions made by the model, 91% of predictions were correct classifications of the brain scan.
- **Precision**: Averaged across each class, out of all positive predictions made by the model for that class, 91% of them were correct classifications of the brain scan.
- **Recall**: Averaged across each class, out of all brain scans that were truly of that class, the model correctly classified them 91% of the time.
- **F1 Score**: The high F1 score indicates a good balance between precision and recall.

Additional Visualizations:



log_roc_before.png

Previous (Logistic) Next (Logistic)

Above we see the ROC curve produced for each class, plotting true positive rate against false positive rate.

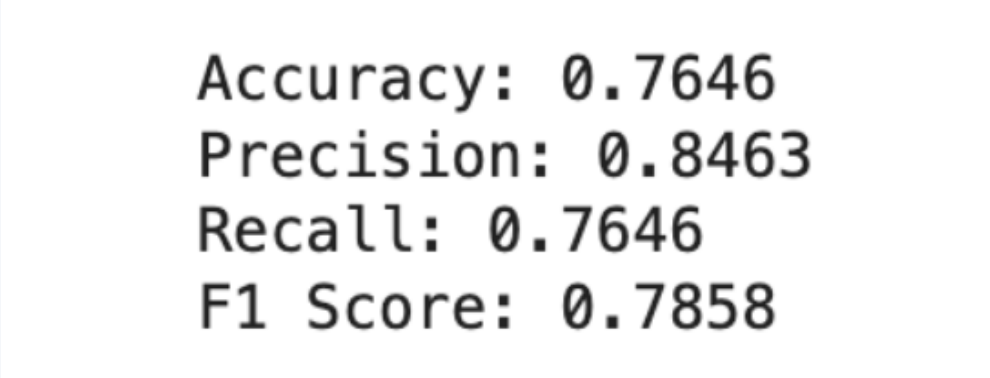
The moderate dementia class had a near perfect area under the curve and the best performance, suggesting that the model was highly effective at identifying patients with moderate dementia. This is evident in the confusion matrix, where we see that there were no false negatives when examining moderately demented patients. For the other classes, they show poorer but still fair performance, with mildly demented patients with the second highest area under the curve followed by non-demented patients then finally patients with very mild dementia.

From the results, it is evident that the model is more effective at identifying certain stages of dementia than others in a pattern very similar to what we see in the logistic regression model. Based on the AOC values, the model can identify moderate and mild dementia relatively well with high sensitivity and specificity. It performs significantly worse with non-demented and very mildly demented patients, which is evident in the confusion matrix as the very mildly demented and nondemented cells show significant mispredictions; the model frequently classified nondemented patients as very mildly demented and very mildly demented patients as nondemented. This pattern makes sense, and similarly to the logistic regression model, we theorize that the pattern is related to magnitude of difference between various classes; non demented and very mildly dementia likely have a very low magnitude of difference between images while mild dementia and moderate dementia are both more distinct, resulting in a relatively higher misclassification rate for nondemented and very mildly demented compared to mild and moderate dementia. The model is likely under fitted for non demented and very mildly demented classes.

Note that this model was trained on a slightly modified dataset, the reasons for which we discuss in the next section.

SVM Results & Discussion

We chose to implement SVM because it can handle high dimension data, and separate different classes with clear margins. SVMs are also less prone to overfitting than neural networks, so we wanted to include the model in our project. Our SVM model shows the following performance metrics when executed on a 20% testing dataset of images:



SVM Performance Metrics

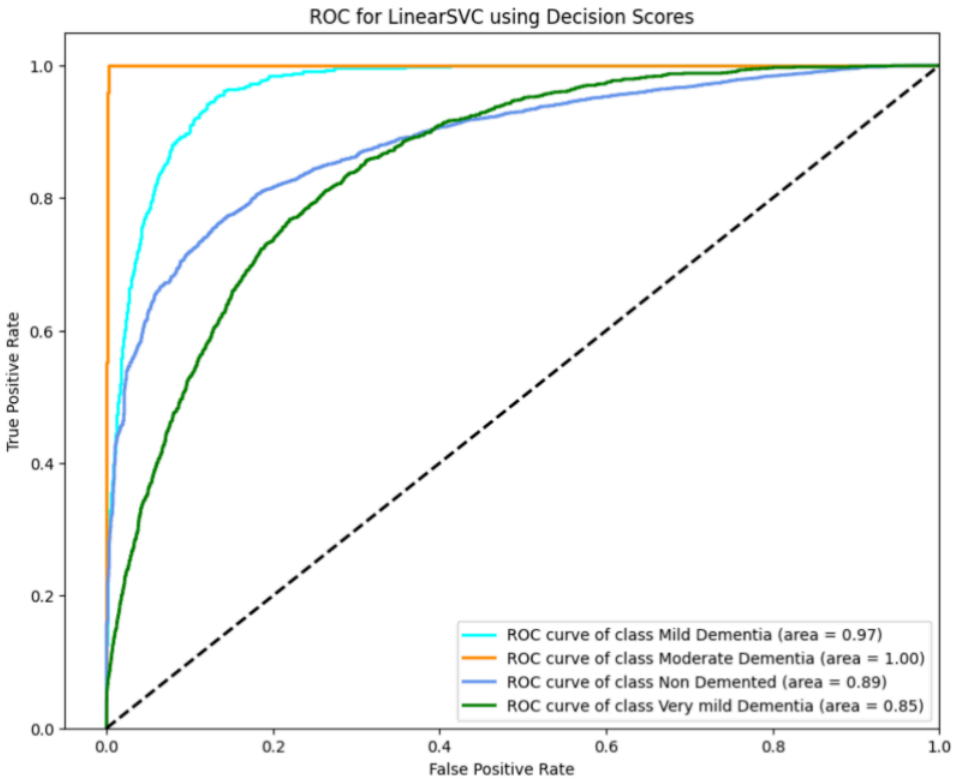
Performance Metrics:

Metric	Value
Accuracy	0.7646
Precision	0.8463
Recall	0.7646
F1 Score	0.7858

We see from the metrics that the model achieved a fair performance on the test data, with values hovering between 0.75 and 0.85 across all metrics. The interpretations of each metric are as follows:

- **Accuracy**: Out of all predictions made by the model, 76.46% of predictions were correct classifications of the brain scan.
- **Precision**: Averaged across each class, out of all positive predictions made by the model for that class, 84.63% of them were correct classifications of the brain scan.
- **Recall**: Averaged across each class, out of all brain scans that were truly of that class, the model correctly classified them 76.46% of the time.
- **F1 Score**: The moderately high F1 score indicates a relatively good balance between precision and recall.

Additional Visualizations:



svm_roc.png

Previous (SVM) Next (SVM)

Above we see the ROC curve produced for each class, plotting true positive rate against false positive rate.

The moderate dementia class had a near perfect area under the curve and the best performance, suggesting that the model was highly effective at identifying patients with moderate dementia. This is evident in the confusion matrix, where we see that there were no false negatives when examining moderately demented patients. For the other classes, they show poorer but still fair performance, with mildly demented patients with the second highest area under the curve followed by non-demented patients then finally patients with very mild dementia.

From the results, it is evident that the model is more effective at identifying certain stages of dementia than others in a pattern very similar to what we see in the logistic regression model. Based on the AOC values, the model can identify moderate and mild dementia relatively well with high sensitivity and specificity. It performs significantly worse with non-demented and very mildly demented patients, which is evident in the confusion matrix as the very mildly demented and nondemented cells show significant mispredictions; the model frequently classified nondemented patients as very mildly demented and very mildly demented patients as nondemented. This pattern makes sense, and similarly to the logistic regression model, we theorize that the pattern is related to magnitude of difference between various classes; non demented and very mildly dementia likely have a very low magnitude of difference between images while mild dementia and moderate dementia are both more distinct, resulting in a relatively higher misclassification rate for nondemented and very mildly demented compared to mild and moderate dementia. The model is likely under fitted for non demented and very mildly demented classes.

Note that this model was trained on a slightly modified dataset, the reasons for which we discuss in the next section.

Comparison and an Aside on Error

Before continuing with our analysis, we would like to note that after obtaining results for the CNN, we sought explanations for the causes of the highly accurate performance. We investigated data leakage as a possible cause, but we see from the image divisions below that there was no leakage.

```
Validation samples: 8644
Testing samples: 17288
Train class distribution: {0: 3502, 1: 341, 2: 47055, 3: 9607}
Validation class distribution: {0: 500, 1: 49, 2: 6722, 3: 1373}
Test class distribution: {0: 1000, 1: 98, 2: 13445, 3: 2745}
Overlap between train and validation: 0
Overlap between train and test: 0
Overlap between validation and test: 0
```

Leakage Investigation

We noticed that the dataset was highly skewed toward nondemented images, with the class containing 137 times more images than the smallest class. This would explain the highly accurate performance, wherein the other classes are “drowned out” and the model is able to accurately predict most of the time. Hence, to control for this bias, we decided to train and test logistic regression and SVM on a smaller subset of the largest class to reduce the disparity. This had a notable effect on the final performance of both logistic regression and SVM. Thus, from this point on, we will compare logistic regression and SVM more directly since they were both trained on balanced datasets, and we will consider CNNs separately.

Overall, based purely on metrics, our initial prediction that CNN would perform the best was correct, followed by logistic regression, then SVM. Comparing the CNN with the other models, the CNN had significantly better performance, encroaching on nearly perfect accuracy. Even with the CNN being trained on a skewed dataset that boosted its accuracy, the degree to which it approached perfect reflects the strengths of CNNs’ learning mechanisms for image classification relative to other models. CNNs use hidden layers and are better at capturing complex patterns, details, and features, especially when it comes to classifying images that are filled with intricate details. Hence, this magnitude of difference between the CNN model and the other models makes sense.

Comparing the logistic regression and SVM, which were both trained on the balanced dataset, the logistic regression model had significantly better performance, encroaching on accuracy nearing 0.9 compared to 0.75 for the SVM mode. The poorer performance of the SVM was likely due to the use of a linear kernel that kept the data points linearly inseparable, producing a greater degree of misclassification. Additionally, both models had lower areas under the ROC curve for nondemented and very mild dementia classes, indicating that they both had the same problem with misclassifying very mild dementia as nondemented and nondemented as very mild dementia. This provides further evidence to support our theory that very mild dementia and nondemented images have very similar features.

As expected, the SVM and logistic regression models performed worse than the CNN model due to being trained on realistic, balanced data sets; however, their performance is not necessarily poor, with metrics hovering around 0.93 for the logistic regression model and 0.75-0.85 for the SVM model. When used to predict dementia on another separate dataset, we expect SVM and logistic regression to retain a greater degree of performance compared to the CNN. This is because being trained on a skewed dataset, CNN is less generalizable and is likely to experience a larger decline in performance when predicting on fresh testing data, whereas SVM and logistic regression will be more robust. If all three models were trained on the same balanced dataset, we hypothesize that CNNs would likely perform better than both logistic regression and SVM due to the levels of feature complexity learning that CNNs can achieve.

Next Steps

We realized while implementing the SVM model that the near perfect performance of the first two models was likely a result of the skewed dataset we were working with. As we mentioned above, this was not corrected for in the CNN but was in the logistic regression and SVM model. In the future, it may be valuable to retrain our model on a completely separate, less skewed dataset and compare its performance to this dataset. We expect that metrics will decline for CNNs and logistic regression, but it will be much more realistic and accurate in practice. We also ran into problems with our computer performance not being able to handle the complexity of some models, specifically the SVM model. We ended up using SVC instead of SVM which definitely could have affected some performance. In the future, we can

invest/borrow the school's more powerful computers to train the model that will definitely be able to handle the large datasets.

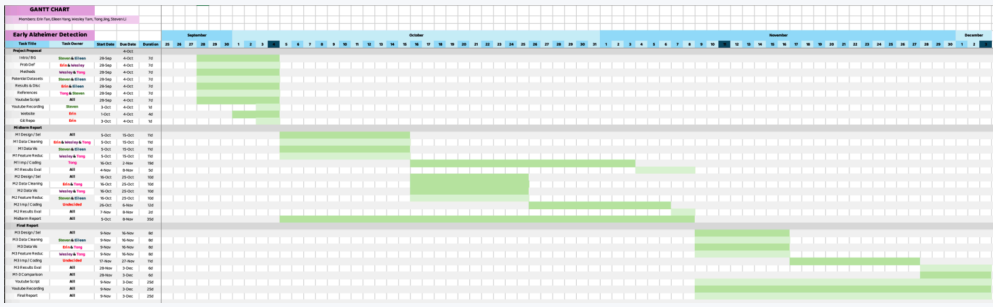
References

[1] A. Bhandarkar et al., "Deep learning based computer aided diagnosis of Alzheimer's disease: A snapshot of last 5 years, gaps, and future directions," Artificial Intelligence Review, vol. 57, no. 2, Feb. 2024. doi:10.1007/s10462-023-10644-8

[2] A. Borji, A. Seifi, and T. H. Hejazi, "An efficient method for detection of Alzheimer's disease using high-dimensional PET scan images," Intelligent Decision Technologies, vol. 17, no. 3, pp. 729–749, Jul. 2023. doi:10.3233/idt-220315

[3] H.-P. Chan, R. K. Samala, L. M. Hadjiiski, and C. Zhou, "Deep learning in medical image analysis," Advances in Experimental Medicine and Biology, pp. 3–21, 2020. doi:10.1007/978-3-030-33128-3_1

Gantt Chart



Gantt Chart

Contributions

Final Contributions

Team Member	Contribution
Erin Tan	• Managed Website • SVM Preprocessing, Model, and Visualization • SVM Analysis • YouTube Script/Slides
Eileen Yang	• Data Visualization • Results & Discussions • Overall Comparison • YouTube Script/Slides/Video
Wesley Tam	• Logistic Regression Preprocessing and Model
Tong Jing	• Gradcam Implementation

Team Member	Contribution
Steven Li	• Data Visualization • Results and Discussions • Overall Comparison • YouTube Script/Slides

Midpoint Contributions

Team Member	Contribution
Erin Tan	• Managed Website • Preprocessing Implementation • Method Analysis
Eileen Yang	• Data Visualization • Results and Discussions • Method Analysis
Wesley Tam	• Preprocessing Implementation
Tong Jing	• Environment Setup • Preprocessing Implementation • CNN Implementation
Steven Li	• Data Visualization • Results and Discussions • Method Analysis

Proposal Contributions

Team Member	Contribution
Erin Tan	• Managed Website • Problem Definition • Potential Results & Discussions
Eileen Yang	• Introduction & Background • Potential Results & Discussions
Wesley Tam	• Problem Definition • Methods
Tong Jing	• Methods • References
Steven Li	• Introduction & Background • References

GitHub Repository