Predicting Brain Tumor Progressions Using CNN and LSTM Pipeline

DS 440 Capstone Project

Hayden Goodreau
College of IST
Pennsylvania State University
State College PA USA
hmg5393@psu.edu

Ethan Sullivan
College of IST
Pennsylvania State University
State College PA USA
ecs5690@psu.edu

Karthik Kuppala
College of Engineering
Pennsylvania State University
State College PA USA
kak615@psu.com

ABSTRACT

This paper focuses on utilizing deep learning techniques to improve clinical decision making and potentially improve outcomes for patients in a timely fashion. In particular, this research employs deep learning techniques to analyze brain tumors using MRI scans from patients themselves. This paper assists in monitoring tumor trajectories, and has the potentially to contribute to extending the survival rates of patients

Our research aims to examine and improve upon existing techniques to accurately grade brain tumors, as well as incorporating novel techniques to predict when patients can expect their tumor to progress to its next stages. In this study, we propose a deep learning pipeline that integrates a Convolutional Neural Network and Long Short-Term Memory network to analyze spatial features with longitudinal MRI brain scans.

In clinical practice, reliable grading and tracking of tumor growth is still very difficult. Manual MRI scan interpretation takes a lot of time and is exposed to challenges like variability, which may cause inconsistent results or delay treatment decisions. Deep learning offers a strong foundation for automating these kinds of manual jobs by using enormous amounts of imaging data to find patterns that the human eye would miss. To address these issues, we propose a novel pipeline using a CNN-LSTM architecture to combine spatial and temporal data

CCS CONCEPTS

Neural Networks, Feature Selection, Semi-Supervised Learning Settings

KEYWORDS

CNN, LSTM, REMBRANDT, Brain-Tumor-Progression, Tumor Grade, DWT, DCT

ACM Reference:

Hayden Goodreau, Ethan Sullivan and Karthik Kuppala. 2025. Predicting Brain Tumor Progressions Using CNN and LSTM Pipeline: DS 440 Capstone Project.

1 Topic

Brain Tumors develop and progress in an unpredictable fashion, most detection and predictive methods currently involve the clinician to manually check individual patient data which as mentioned raises risks for misinterpretation and is time consuming.

Our novel approach introduced CNN+LSTM pipeline that uses deep learning as its core component to help assist in clinical decision-making via better knowledge of brain tumor development using MRI scans from large datasets employed from the Cancer Imaging Archive. In our model not only will patients Brain Tumors be classified, but also their next grade progression will be predicted, therefore providing doctors with a better picture of what might lie ahead for their patients and enabling them to act early in the course of treatment.

To validate our model architecture, we used a dual-dataset approach to help address the limitations of each dataset individually. The model architecture along with our data approach provides a strong foundation for our research and the ability for future additions.

2 Importance

Deep learning techniques have emerged as an incredibly powerful modeling approach in the medical field, giving medical professionals an upper hand not widely available in the domain of Brain Tumors and MRI scans before.

An emphasized before the nature of Brain Tumors is being unpredictable and aggressive; this presents a significant barrier to clinicians trying to predict next grade tumor progression. Predicting tumor growth accurately is essential for developing individualized treatment plans and enhancing patient outcomes. Present models provide conventional imaging analysis and grading techniques which rely significantly on clinicians subjective and unreliable manual interpretation.

Deep learning presents a viable way to improve and automate this process, especially when using deep learning models like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. By implementing spatial and temporal patterns from MRI sequences, these models can increase diagnosis accuracy while lowering the need for human interpretation.

Our approach tries to fill in this gap by combining temporal tumor growth trends with spatial analysis of MRI images. We hope our strategy may enable physicians to intervene early and better track the course of Brain Tumor Progression.

3 Prior Works

There were two main aspects that our model aims to achieve. One aspect (our novel approach) involves using an LSTM model to accurately predict when patients can expect tumors to move to the next grading stage. This approach was not readily available in prior works, and we largely approached that portion of the model from scratch. We did however draw on existing works on in our own research to learn some of the best methods to train our model to grade tumors. We ultimately utilized two research papers, one that involved a CNN model, and one that used an LSTM model.

3.1 Prior CNN Model Research

The first paper we utilized, "*Brain Tumor Classification Using Convolutional Neural Networks (Seetha, Raja)",* presented a model that automatically classified brain tumors with MRI sequences using a CNN model. This paper highlighted that it was able to outperform some of the more conventional machine learning methods, achieving a training accuracy of 97.5%, accompanied with high validation accuracy and low loss metrics. This paper utilized two public datasets, including BRATS 2015 and Radiopaedia. While this model performed quite successfully, the authors did acknowledge that there were difficulties processing and transforming such a large quantity of MRI data, indicating that these performances took place using a subset of the data.

Prior Works Algorithm for CNN Based Classification

1. Apply convolution filter in first layer.
2. Smooth sensitivity of convolution filter.
3. Activation layer controls signals from one layer to the next.
4. Training period is fastened using RELU
5. Neurons in the proceeding layer are connected to neurons in the subsequent layer.
6. Loss layers are added during training to provide feedback to the neural network.

3.2 Prior LSTM Model Research

The second paper that our research team utilized, *"Brain tumor grade classification Using LSTM Neural Networks with Domain Pre-Transforms (Fasihi, Mikhael)",* uses a weakly supervised learning approach to classify brain tumor grades on MRI images. These authors utilized discrete wavelet transformations and discrete cosine transforms to capture spatial and frequency characteristics of medical images. This study worked with the same dataset that our team used, REMBRANDT, and so the results of this paper were especially useful to us, giving us a fantastic picture of what had already been achieved using this dataset.

This model was able to combine both transforms to achieve a classification accuracy of 86.98%, outperforming their previous models. While their model is considered quite successful, the authors did note that there were existing challenges of limited high-quality medical images that could be leveraged in their model.

Prior Works Algorithm for LSTM Classification

1. Preprocess images using ROI extractions.
2. Feature Extraction using DCT and Wavelets transformations.
3. Extract feature vector information.
4. Employ neural network classification.
5. Evaluate and test for accuracy

4 Design

4.1 Data Collection

To train a predictive model for tumor grade progression, we required two features:

1. Patients with multiple timestamps to determine progressive tumors.

2. Patients with assigned tumor grades.

An early challenge was finding publicly available MRI images, let alone datasets that contain the features above. We were able to obtain clinical MRI images from The Cancer Imaging Archive after applying for access to the following restricted datasets: REMBRANDT and BRAIN-TUMOR-PROGRESSION

REMBRANDT consists of 130 patients and 110,020 MRI images, complete with MRI sequence type, disease name, and tumor grades. While this dataset is sufficient for training a predictive model for image classification, it lacks the crucial element of multiple timestamps. To address this, we used BRAIN-TUMOR-PROGRESSION to fill in the gaps.

BRAIN-TUMOR-PROGRESSION (BTP) is considerably smaller than REMBRANDT, with 20 patients and 8,798 MRI images. The benefit of the BTP dataset is the inclusion timestamps, with each patient consisting of two separate MRI imaging dates several months apart. The two imaging dates allows us to determine which patients exhibit tumor growth, ideal for learning spatial features of progressive tumors.

4.2 Data Cleaning

Another small challenge our team experienced came with the consistency within our datasets. Even though the majority of patients within the REMBRANDT dataset contained grades, some did not. This required manual data cleaning, as our team had to use OS code functions to discover which patient folders contained patient grades within their metadata. If the patient did not contain the necessary grades, we manually removed those patient folders from our dataset.

4.3 Data Integration

While we could not identify a single data source that contains multiple imaging dates and tumor grades, by utilizing the vast library of REMBRANDT, our team set out to create a classifier to assign pseudo-labels to the BTP patients. Through this method, we obtained the ideal dataset to train a classifier tasked for brain tumor progression prediction.

We recognized that the lack of ground-truth labels for the BTP patients introduces room for incorrect predictions and assumptions. To mitigate risk, we sought to create a strong classifier using a Convolutional Neural Network (CNN) architecture. Known for their superior power in analyzing images, CNN is ideal to fill in the gaps of our BTP data.

To properly learn the spatial features and patterns of MRI sequences, we sought to utilize a Long Short Term Memory (LSTM) architecture. By using entire MRI sequences as input, the LSTM model will be able to capture information over multiple sequences, learning the features that indicate growth patterns in tumors. To allow the LSTM to analyze the MRI sequences, we processed sequences and represented them as feature vectors of length 100.

Appending the grade labels, dates, and feature vectors to the patients of BTP provides us with the ideal data frame to perform brain tumor progression prediction. Figure 1 demonstrates the data integration pipeline.

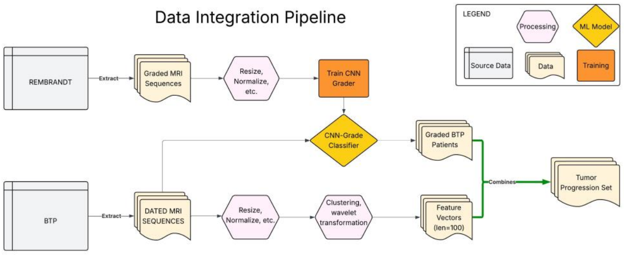


Figure 1: Data Integration. A combination of CNN trained labels with custom processed feature vectors provides sufficient information for brain tumor grade progression prediction.

5 Image Processing

5.1 CNN Processing

Our team began preprocessing images within our CNN model by resizing each image to 224 x 244 pixels. This was due to the fact that the MRI images within the REMBRANDT dataset were not sized in a uniform manner. We then converted these images into an RGB format using PIL, which is standard transformation practices that are required when using a CNN model. Images were also normalized using mean and standard deviation values of [0.485, 456, 406] and [0.229, 0.224, 0.225], which is also considered typical output values. Lastly, these images were converted to tensors, which effectively converted our grayscale images that came in forms of 2D tensors (height x width) and added a channel dimension that was used when inputting all images into our CNN model.

Our preprocessing searched recursively within the REMBRANDT dataset, finding all DICOM files and shuffling them for randomization purposes. After this is completed, our model retrieved patient IDs from the DICOM metadata to match accordingly at a later point in our pipeline.

Due to tensor conversions, loading images became very computationally expensive, and we were unable to load the entire dataset at once. This forced us to adapt our preprocessing and data loading methods, instead loading in batches of one-thousand images and processing them sequentially. We loaded ten batches at a time, shuffling to ensure randomization, and at a later point in our pipeline iteratively loaded in new image batches to continuously train our model on.

5.2 LSTM MRI Sequence Processing

To represent the MRI sequences as feature vectors, we mirrored the processing techniques found in [3]. Each patient was split into patient-date pairs and treated as different individuals. For example, patient PGBM-001, who had imaging dates on ‘04-02-1992' and ‘11-19-1991', would be represented as ‘PGM-001_04-02-1992' and ‘PGM-001_11-19-1991'. Rescaling, normalizing pixel values between 0 and 1, and resizing to 256 x 256 was performed on each MRI image associated to the patient-date pairs. Afterwards, clustering was used to find the ROI for the MRI images. Tumors contrast against the dark backdrop of the brain in an MRI image. By grouping the image into clusters of three based on pixel intensity, feature extraction can be performed on the tumor region of the brain.

Upon determining the ROI, the image is further broken down four matrices representing pixel intensity sub-bands:

- LL: A low frequency approximation of the overall image, capturing general shape and intensity trends
- LH: Highlighting vertical edges in high-frequency components from the top and bottom of the image
- HL: Highlighting vertical edges in high frequency components from the left and right of the image

- HH: Highlighting diagonal edges and finer texture details in high-frequency components.

After capturing pixel frequency changes in these matrices, further processing is performed via Discrete Cosine Transform (DCT) on the LL band, recording the most significant frequency changes. These transformations are performed on each image in an MRI sequence, recording key spatial information from a sequence into a vector.

6 Training/Testing CNN

6.1 Creating a Custom Data Class

Upon completion of our preprocessing and image loading steps, our group defined a custom dataset class in order to properly manage our preprocessed batches. This custom dataset saved the batches as tensor files that corresponded to the correct patient identification contained in our metadata. This allowed our data to effectively be loaded into our PyTorch model using a data loader

Within this custom data class, we continued to use best data preparation practices, ensuring that there were no remaining overlooked elements from our manual data cleaning and dropping rows where grades were missing to ensure that our training model only utilized properly labeled data. We then converted the tumor grades into explicit integers and created a label map to map image tensors with the appropriate labels further down our pipeline. This class then loads tensor batches from the directory, checking if each patient ID has the correct label. If it does, the pair are stored in the list that will be used for training.

6.2 Model Building

Our team utilized a CNN model based on ResNet-18 architecture, which is pretrained on ImageNet and is ideal for image classification tasks such as this. We utilized the pretrained weights to ensure that our model is provided with a great foundation for feature extraction and pattern learning.

We customized this imported model by replacing the final fully connected layer, switching the original one-thousand classes (based on ImageNet classes) to three, which is much better suited to our tumor grade classes, allowing for multi-class classification of tumor grades and progression.

6.3 Model Training/Testing

After setting hyperparameters, the preprocessed image tensor that were now passed through ResNet-18 were loaded into the model, and class weights were computed to address any imbalances in how the labels were distributed. The weights were then passed to an entropy loss function to penalize any underrepresented classes (which we had within our dataset) or misclassifications. We then split the dataset into training, validation, and testing sets (80%, 10%, 10%), and then data loader objects were created to handle the prior batching and shuffling.

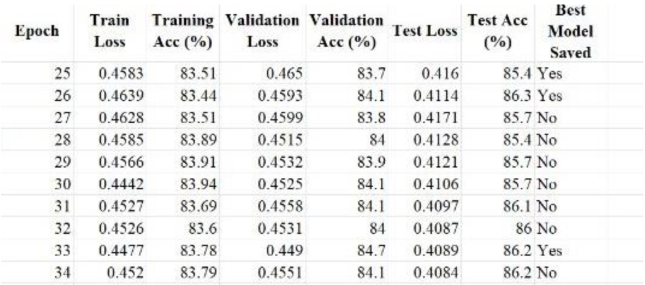
We then create a best path model checkpoint to start training from scratch and then save the best model that was found from the training as the new checkpoint to resume training from.

Our training loop iterated over ten epochs per run, moving forward with the standard training steps that come with a CNN model, including forward propagation, zeroing gradients, computing loss, optimization, and backward propagation. We also included an Adam optimizer that worked in conjunction with a learning rate scheduler. The Adam optimizer adjusts the learning rate throughout training, helping to optimize our model, while the learning rate scheduler decreased the learning rate slightly after every five epochs in order to decrease the chances of overfitting our model to the training data and allow our model to converge efficiently and effectively.

After each epoch, we evaluated the model on testing and validation results, calculating loss and classification accuracies. If the validation accuracy of our newest epoch surpassed the previous best model checkpoint, the newest epoch was then saved as the new best model. This made sure that we were consistently utilizing the model that generalized best towards new and unseen data.

7 Results CNN Grading

Our training and testing results displayed consistent performance and improvement as we continued to load and train our model on new batches of ten thousand images. The table below displays our group's best classification accuracy results. This model achieved a best-validation-accuracy score of 84.7%, and this was the model that was saved to our best model checkpoint. At this epoch, training and testing accuracy resulted in 83.78% and 86.2% validation accuracy, respectively, indicating that the model performs similarly on all three datasets. This also indicates that our model is not overfit to this particular batch of data, which makes sense considering that this model was run on new image batches multiple times. To go along with these classification accuracy metrics, training, validation, and testing loss metrics remained stable as well (0.4477, 0.449, 0.4089), further cementing the point that our model generalizes well on unseen data. These results reflect a model that has been well-tuned over time. Our accuracy metrics were able to come quite close to matching the validation accuracy of our prior works that trained on the REMBRANDT dataset (86%).



Epoch	Train Loss	Training Acc (%)	Validation Loss	Validation Acc (%)	Test Loss	Test Acc (%)	Best Model Saved
25	0.4583	83.51	0.465	83.7	0.416	85.4	Yes
26	0.4639	83.44	0.4593	84.1	0.4114	86.3	Yes
27	0.4628	83.51	0.4599	83.8	0.4171	85.7	No
28	0.4585	83.89	0.4515	84	0.4128	85.4	No
29	0.4566	83.91	0.4532	83.9	0.4121	85.7	No
30	0.4442	83.94	0.4525	84.1	0.4106	85.7	No
31	0.4527	83.69	0.4558	84.1	0.4097	86.1	No
32	0.4526	83.6	0.4531	84	0.4087	86	No
33	0.4477	83.78	0.449	84.7	0.4089	86.2	Yes
34	0.452	83.79	0.4551	84.1	0.4084	86.2	No

Figure 2: CNN Testing Results. Displays training, testing, and validation accuracy and loss metrics across our best batch of epochs. Also displays which epochs were saved to best model checkpoint.

8 BTP Grading With CNN

While we were quite pleased with the results of our grading classifications using a CNN, our top priority was to build an accurate model that could accurately classify tumor grades so that our team could overcome the lack of grades available in the Brain-Tumor-Progression dataset. Having built an effective model, the next step in our pipeline was to effectively assign grades to each patient within the BTP dataset in order to carry out our novel approach of predicting the number of months it takes for each patient's tumor to progress to its' next stage.

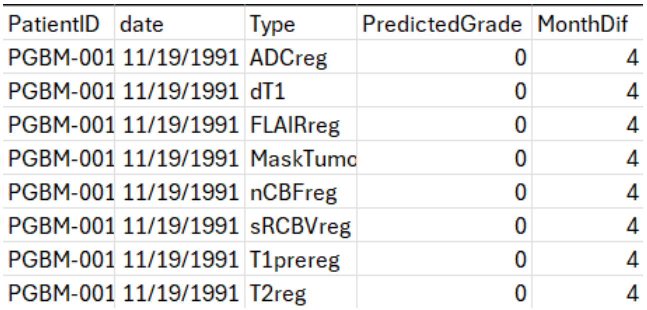
We created a transformation pipeline that resized images, converted to tensors, and normalized using the same specifications that were used for images coming from REMBRANDT. We then loaded the model saved from our best model checkpoint, and changed our model to disable gradient computations, setting for evaluation instead. Once again, each image is converted to a PIL RGB image and then passed through the model. Each prediction is collected, and once all predictions are collected, our function returns the most common grade predicted for each scan.

In order to obtain the best possible results, we wanted to analyze the folders that contained MRI scans of the entire brain holistically. Some folders contained MRI scans that included bisections of the brain and didn't necessarily contain the region of the brain with the tumor. The best brain scans for our process were T1Post and FLAIR scans. These scan types include an overhead view of the whole brain and proved most effective for our model to assign tumor grades to each patient. All predictions were saved into a CSV file that was utilized by our LSTM model to carry our novel approach.

9 LSTM Training/Testing

9.1 Data set-up

Following the results of the CNN grading, analysis was necessary to eligible patients for training. Patients who showed positive tumor grade progression between the two sequencing dates (grade 2 to 3, 4) were eligible. Out of the 20 patients in the BTP dataset, only 6 showed positive tumor grade progression.



PatientID	date	Type	PredictedGrade	MonthDif
PGBM-001	11/19/1991	ADCreg	0	4
PGBM-001	11/19/1991	dT1	0	4
PGBM-001	11/19/1991	FLAIRreg	0	4
PGBM-001	11/19/1991	MaskTumo	0	4
PGBM-001	11/19/1991	nCBFreg	0	4
PGBM-001	11/19/1991	sRCBVreg	0	4
PGBM-001	11/19/1991	T1prereg	0	4
PGBM-001	11/19/1991	T2reg	0	4

Figure 3: Patient reference sheet utilized to prepare the tumor grade progression data

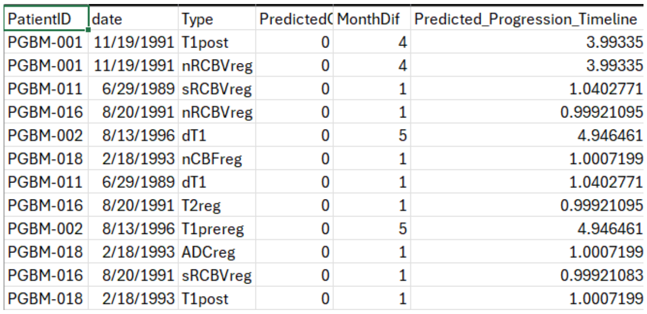
A reference csv was created to collect the PatientID, date, Type of MRI sequence, and the CNN predicted grade. 'MonthDif' is a calculated field showing the difference in time between the sequencing dates in months for the patient. This is the target value that the LSTM model will be tasked with predicting. This information was combined with the feature vectors created in section 5.2 to complete the brain tumor grade progression data set.

9.2 Architecture

For our task, we utilized a Bidirectional LSTM for regression, with 128 hidden dimensions. The bidirectional nature is used to capture temporal patterns across the sequences, and 1 layer is used due to the limited size of the input data. The output is a single regression value indicating the number of months until the tumor will progress to the next grade level. 30 Epochs are used to train the model, using early stopping to avoid overfitting. A minimum validation loss improvement of 0.01 over 5 epochs is required before early stopping is triggered. The model was trained on 48 patient-date-sequence pairs and tested on 12 patient-date-sequence pairs.

10 LSTM Results

Initial results show strong accuracy, with a test Mean Absolute Error of 0.02 months. Predicting the value 'Predicted_Progression_Timeline', the results are nearly an exact match to their ground truth labels. While the train loss vs validation loss in **Figure 5** appears to show a generalizable model, the small amount of training material is likely resulting in overfitting, despite the use of early stopping.



PatientID	date	Type	Predicted(	MonthDif	Predicted_Progression_Timeline
PGBM-001	11/19/1991	T1post	0	4	3.99335
PGBM-001	11/19/1991	nRCBVreg	0	4	3.99335
PGBM-011	6/29/1989	sRCBVreg	0	1	1.0402771
PGBM-016	8/20/1991	nRCBVreg	0	1	0.99921095
PGBM-002	8/13/1996	dT1	0	5	4.946461
PGBM-018	2/18/1993	nCBFreg	0	1	1.0007199
PGBM-011	6/29/1989	dT1	0	1	1.0402771
PGBM-016	8/20/1991	T2reg	0	1	0.99921095
PGBM-002	8/13/1996	T1prereg	0	5	4.946461
PGBM-018	2/18/1993	ADCreg	0	1	1.0007199
PGBM-016	8/20/1991	sRCBVreg	0	1	0.99921083
PGBM-018	2/18/1993	T1post	0	1	1.0007199

Figure 4: Predicted Progression Timeline shows accurate month-until-progression predictions. With a small training set, overfitting is a likely consequence

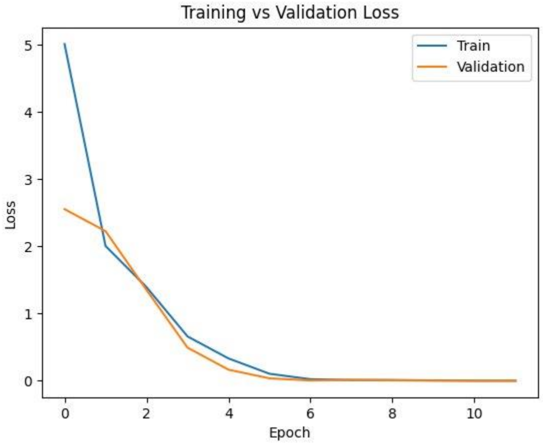


Figure 5: Training vs Validation Loss shows a healthy relationship during training.

11 Discussion

This model pipeline provided some valuable and promising insights into using multi modal deep learning techniques to carry out segmentation predictions and tumor classifications. We were encouraged by the performance of this model, and we feel that these results provide an excellent foundation for future works.

We did encounter a few challenges throughout our process, most of which came from data processing and loading issues. As mentioned previously, processing and converting MRI images in tensor format proved quite computationally expensive, and we did not have the tools necessary to process the entire REMBRANDT dataset all at once. This forced us to adjust our model and load in data in batches, retraining our model with new random batches. Once data was loaded in, we found runtime to be an issue as well. Training thousands of images at a time increased the training runtime to several hours per train, slowing down our training process as a whole.

Another issue that our team encountered was a slight data imbalance in our REMBRANDT dataset, as that dataset comparatively lacked in stage 3 tumors. This likely led to some incorrect predictions when our CNN model was used to assign grades to our Brain-Tumor-Progression dataset. We noticed that our grading outputs within our BTP CSV lacked stage 3 grades, and data imbalance is the likely cause.

Finding the necessary datasets with the qualifications that we were looking for proved to be somewhat of a challenge. Some data was too small. Others were not publicly available. Still, some others lacked date timestamps or the necessary tumor grades. We overcame this issue using multiple datasets and adaptation of our pipeline, but in the future, we are looking forward to the development of new datasets that are even more robust. This will almost certainly lead to even more promising results.

While we did encounter some challenges that hindered the progress of our model at times, we were largely able to overcome these setbacks and achieve promising results. Our classification accuracy was very encouraging, and we were able to identify enough patients within the BTP dataset to provide progression prediction metrics that are a great foundation for future work with more robust data.

12 Conclusions

This team was able to overcome a lack of available competent data through the utilization of multiple datasets and semi-supervised learning techniques. Pipelines such as this have proven to be a consistent method for addressing any data incompetencies and pushing forwards into novel territory, which in this case involves moving past tumor classifications and incorporating tumor progression into the model as well. We hope that results such as this can serve as a baseline for future works looking to address similar inconsistencies.

Our results demonstrate that next-grade tumor progression prediction is achievable using LSTM architectures. Furthermore, data integration techniques and pseudo-labeling can adequately address data impurities. Our proposed pipeline is capable of both grading tumors and projecting progression timelines, presenting both patients and doctors valuable insights for medical treatment planning.

Despite dataset limitations and hardware constraints, our architecture performed well on both classification and regression tasks. The use of deep learning techniques like CNN-grade prediction compensated for the lack of publicly available and annotated data, enabling the construction of robust architectures that lay the groundwork for temporal grading predictions.

ACKNOWLEDGMENTS

We thank The Cancer Imaging Archive (TCIA) for providing speedy access to the REMBRANDT and BRAIN-TUMOR-PROGRESSION datasets. We would also like to thank Professor

Jinghui Chen and Tianrong Zhang for their valuable insights and guidance throughout the course of the semester.

REFERENCES

[1] M. S. Fasihi and W. B. Mikhael, "Brain tumor grade classification Using LSTM Neural Networks with Domain Pre-Transforms," 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Lansing, MI, USA, 2021, pp. 529-532, doi: 10.1109/MWSCAS47672.2021.9531760. keywords: {Training;Wavelet domain;Handheld computers;Computer architecture;Transforms;Discrete wavelet transforms;Discrete cosine transforms;Brain tumors classification;Medical image classification;Machine Learning;Deep Learning}

[2] Seetha J, Raja S. S. Brain Tumor Classification Using Convolutional Neural Networks. Biomed Pharmacol J 2018;11(3).

[3] Scarpace, L., Flanders, A. E., Jain, R., Mikkelsen, T., & Andrews, D. W. (2019). **Data From REMBRANDT [Data set].** The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2015.588OZUZB

[4] Schmainda KM, Prah M (2018). **Data from Brain-Tumor-Progression**. The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2018.15quzvnb