From Data to Diagnosis: Detecting Gliomas with Deep Learning

Author: Shannon Werner

Introduction

Gliomas are a class of brain tumors that vary widely in aggressiveness, and early detection is critical for determining the correct treatment and improving prognosis. A current shortage of radiologists, however, makes this a challenging undertaking. Without significant changes, it is nearly impossible, in fact. One reason for the shortage is an aging population that requires more imaging and is growing disproportionately to the number of radiology trainees. Between 2010 and 2020, the number of adults aged 65 and older grew by 34%, while the number of trainees increased by only 2.5% (Medicus Healthcare Solutions, 2025). Automating some of the work of radiologists could help with this shortage and make radiologists more efficient overall.

This project contributes to that automation by developing and evaluating a machine learning model that can accurately classify brain MRI images into three categories: normal (no glioma), low-grade glioma (grade 1/2), and high-grade glioma (grade 3/4) (Brain Tumour Research, n.d.). Such a model would alleviate some of the workload of radiologists and allow doctors to catch tumors sooner, which would lead to better prognosis and survival rates. A Deep Convolutional Generative Adversarial Network (DCGAN) was created and used for data augmentation to double the size of the dataset. Simple Framework for Contrastive Learning Visual Representations (SimCLR) was then used to learn visual representations in order to help with the downstream task of classification. Three different deep-learning pretrained classification models were used and compared; Convolutional Neural Network (CNN), Multi-Layer Perceptron Mixer (MLP-Mixer), and Visual Transformer (ViT).

There are several personal motivations for choosing this project. Out of all the different applications of AI that I have learned about, cancer detection has been one of the most interesting to me. I appreciate how profoundly it can help people and to have the opportunity to understand the theory behind it was a powerful way to apply my skills to something that truly matters to me. I specifically chose to work with brain MRIs because my father was a neuroradiologist and this project felt like a meaningful way to connect with the field that he dedicated his life to. Additionally, a close family friend passed away from the most aggressive type of glioma, a glioblastoma, which gave me even more drive to work on these early detection tools.

Related Work

Several prior studies helped provide important context for the deep learning approaches used in this project. Each study shares similarities with the current work, either through the classification task, the types of unsupervised methods applied, or the deep learning architectures explored. However, the proposed project combines and extends these ideas by incorporating both generative and self-supervised techniques, and by individually evaluating multiple supervised models to better understand their individual strengths and weaknesses in a medical imaging context.

One related study is directly aligned with the proposed classification task of categorizing images into normal, low-grade glioma, and high-grade glioma (Goceri, Yılmaz, Uysal, Ergen, & Doğan, 2024). The deep learning method used is one that combines CNN and transformer blocks in a hybrid structure and there are no unsupervised techniques. In contrast to the study, in this project, ViT will be explored independently as well as CNN and MLP-Mixer, along with GAN and SimCLR for unsupervised methods.

Another study applies a DCGAN to generate synthetic MRI images and increase the training set for a ViT model (Benz, Ham, Zhang, Karjauv, & Kweon, 2021). It finds that generated tumor images boost ViT accuracy from 86.3% to 99.3%. This project extends that study by additionally using contrastive learning with SimCLR. In addition, instead of focusing solely on ViT, the effectiveness of CNN and MLP-Classifier will be evaluated as well.

A third study utilizes CNNs, MLPs, and ViTs across standard image classification tasks. Rather than evaluate each model individually, the study combines them in an ensemble approach (Wang, Li, Zheng, & Huang, 2022). While this does confirm the value of exploring multiple classifier models, it does not incorporate any unsupervised learning

techniques. Additionally, it focuses on nature image datasets. In contrast, this project applies these three architectures to medical images and integrates DCGAN and SimCLR to enhance performance.

Data Source

Data was collected from three different sources to create a combination of brain MRI images with three different labels. Images for normal brain MRIs were sourced from the IXI dataset (Information eXtraction from Images [IXI] dataset, n.d.). Images for low-grade gliomas and high-grade gliomas were sourced from The Cancer Imaging Archive (TCIA) from UCSF data (UCSF Center for Intelligent Imaging, 2022) and additional images for low-grade gliomas were sourced from Figshare (Parisot et al., 2015). TCIA provided 56 low-grade images and Figshare provided 210 for a total of 266 low-grade images. The same number of high-grade and normal images were randomly selected to obtain a balanced dataset. The Figshare images were FLAIR MRI images and the rest were T1-weighted images. All images were downloaded as NifTI files, which is a standard file format for medical imaging data.

Feature Engineering

All original data went through the same preprocessing steps listed below.

1. **Filter** into one of three lists: normal, low-grade, or high-grade.
2. **Reorient** to the RAS (Right, Anterior, Superior) coordinate system to maintain spatial consistency across all images. In this orientation, the axes increase from left to right along the x-axis, back to front along the y-axis, and bottom to top along the z-axis.
3. **Resample** to ensure all inputs are the same resolution by calculating the required zoom factor for each axis.
4. **Extract** an axial slice so that all views are the same. This is the middle slice along the z-axis. It is the top down view and extracting it converts the data from a 3D image to a 2D representation.
5. **Resize** the 2D slice to a fixed shape, regardless of original dimensions, which is needed for DCGAN.
6. **Normalize** pixel values to [-1, 1] in preparation for the DCGAN, which utilizes the tanh function.

Unsupervised Learning

Methods

GAN

Given the small size of the dataset, a DCGAN model was chosen to increase the number of images in order to improve accuracy of the classifiers. The number of images for each grade was doubled in order to take the size of the dataset from 798 images to 1,596 images. The architecture of the DCGAN was modeled after an example provided in a course assignment, which itself was based on the original design proposed by Radford, Metz, and Chintala (2015).

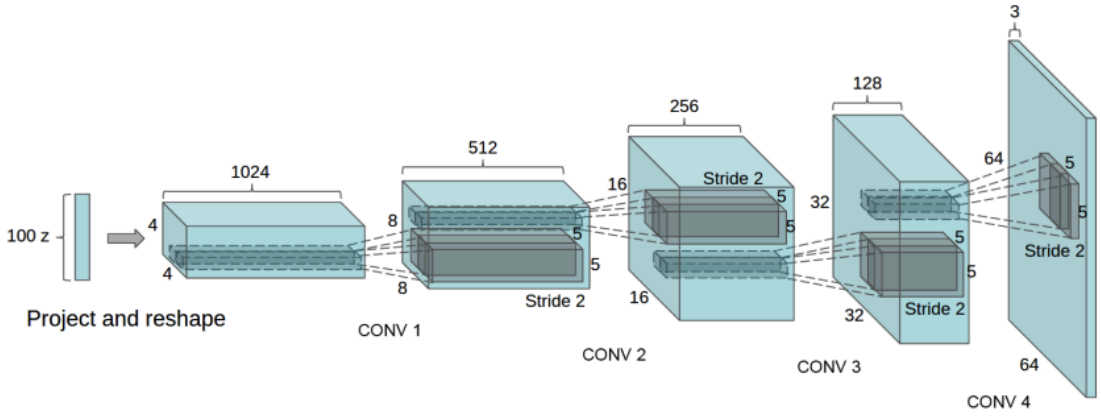


Figure 1: DCGAN generator architecture. A 100-D noise vector is projected and gradually increased in size to a 64×64 image via four fractionally-strided convolutions.

To improve the quality of the generated outputs. I experimented with several different training strategies. Initially, I lowered the learning rate, but then increased it again after seeing little effect. I began tracking the Fréchet Inception Distance (FID) score, which is commonly used to assess the quality of GAN images. As shown in Figure 2, the discriminator loss was decreasing and the generator loss was increasing, which indicated that it was struggling to learn. This can be seen in Figure 2 below. To try to mitigate this, I switched to training the generator twice as often as the discriminator, which improved visual results, but the FID score remained high at ≈ 343. In response, I changed from z-score normalization in the setup to a min/max normalization and the score went down to ≈ 333. However, FID scores above 50 suggest noticeable differences from the real images (ApX Machine Learning, n.d.), so these values indicate that the generated images were still not perceptually convincing.

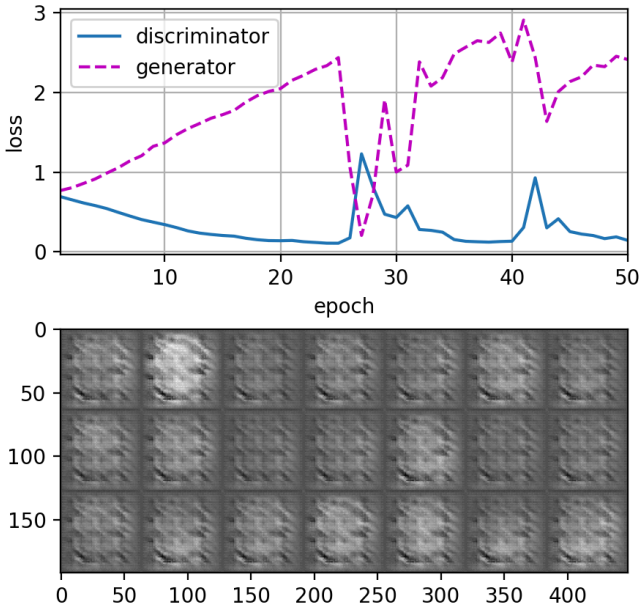


Figure 2: DCGAN training over 50 epochs. Discriminator loss steadily decreases, while generator loss increases and fluctuates, indicating that the generator is struggling to learn as the discriminator becomes stronger. The generated outputs (bottom) show little visual improvement.

I researched the FID score more and learned that it still suffers from a lot of drawbacks, and most notably, it suffers from a high bias (Skandarani, Jodoin, & Lalande, 2023), so I switched to the Learned Perceptual Image Patch Similarity (LPIPS) metric. This measure has been shown to match human perception well (Lightning AI, n.d.) and the results are easy to interpret with similar images having scores around 0.0 to 0.2 (Zhang, Isola, Efros, Shechtman, & Wang, 2018). In the first run, the LPIPS score was ≈ 0.8158. After looking further into how to improve my results, I set bias to False for the Conv2D layers and removed batch normalization in the first layer of the discriminator, as these are common practices (PyTorch, n.d.). These changes significantly improved the LPIPS score to ≈ 0.1214. Different numbers of epochs were also tested throughout, with 300 ultimately producing the best LPIPS score. I started at 50 epochs and slowly increased to give the model more time to learn and improve the quality of the output.

SimCLR

SimCLR was conducted to learn visual representations and improve downstream performance of the classifiers in supervised learning. This approach was particularly effective for this project because it allowed me to leverage a small dataset by simply not including the labels. High-quality representations can be produced when strong augmentations are used and a non-linear projection head is applied during training (Chen, Kornblith, Norouzi, &

Hinton, 2020). After training, the encoder was saved and used as a feature extractor for the supervised learning portion, which helped improve the performance of the pretrained classifiers by capturing more meaningful patterns specific to the data domain.

The original SimCLR proposes the following transforms, but these are used for natural image datasets, such as Imagenet, and would not make sense for medical images.

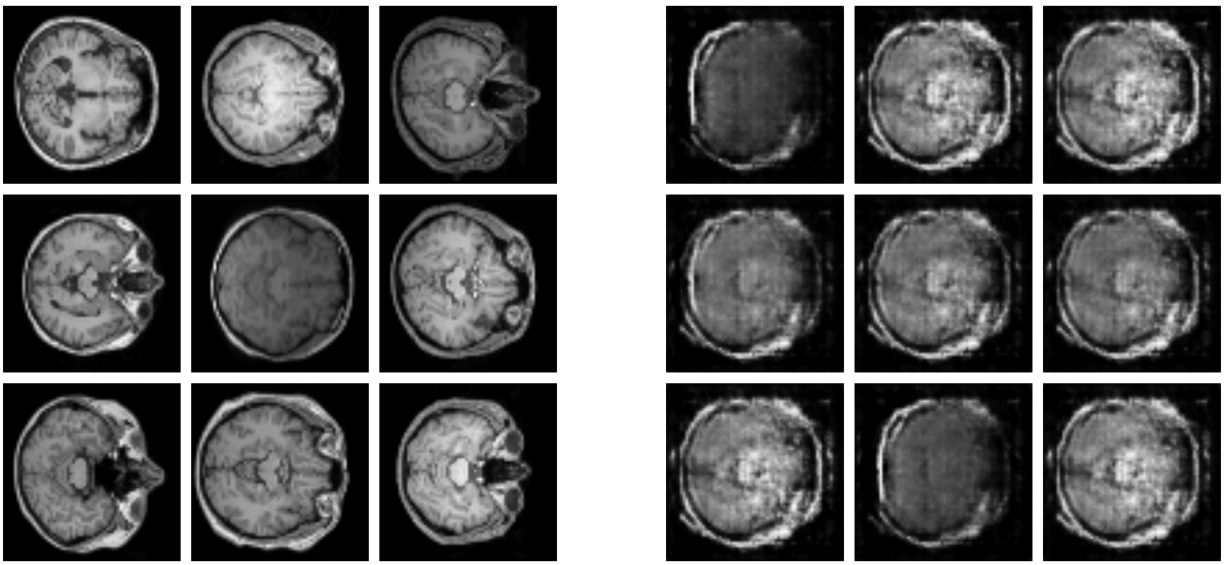
1. transforms.RandomResizedCrop(224, scale = (0.08, 1.0))
2. transforms.RandomHorizontalFlip( )
3. transforms.ColorJitter(0.8, 0.8, 0.8, 0.2)
4. transforms.RandomGrayscale(p = 0.2)
5. transforms.GaussianBlur(kernel_size = 23, sigma = (0.1, 2.0))

Cropping could remove important regions, such as tumors or subtle abnormalities. Flipping may create anatomically implausible structures, such as moving a tumor from the left hemisphere to the right. Color jitter mimics random lighting changes, but in MRIs, pixel intensities represent tissues properties, so altering them could distort meaningful information. To fix these issues, only Gaussian Blur and a minor cropping were included from this list. Random affine transformations were added to bring geometric variety while maintaining spatial structure. The final transform list was as follows:

1. transforms.RandomResizedCrop(224, scale = (0.9, 1.0))
2. transforms.GaussianBlur(kernel_size = 5, sigma = (0.1, 0.5))
3. transforms.RandomAffine(degrees = 5, translate = (0.02, 0.02))

Evaluation

DCGAN



(a) (b)

Figure 3: Comparison of real (a) and DCGAN-generated (b) normal brain MRIs.

The choice of evaluation metric for the DCGAN was the LPIPS score. After tweaking hyperparameters and achieving an excellent score with one grade of images, the rest of grade images were put through the DCGAN. The learning rate was increased for the low-grade and high-grade images, as they vary more in appearance and can benefit from such a switch. While all the other datasets achieved scores of under 0.13, the low-grade images initially

scored ≈ 0.2314. I split that dataset into two based on which dataset the images originally came from and the scores lowered to be consistent with the rest. The final results are listed below.

Image Grade	Learning Rate	Number of Epochs	Best LPIPS Score	Epoch of Best Score
Normal	0.0002	300	0.1123	290
Low-grade	0.0005	300	0.1227	300
Low-grade (UCSF)	0.0005	300	0.0883	280
High-grade	0.0005	300	0.1048	270

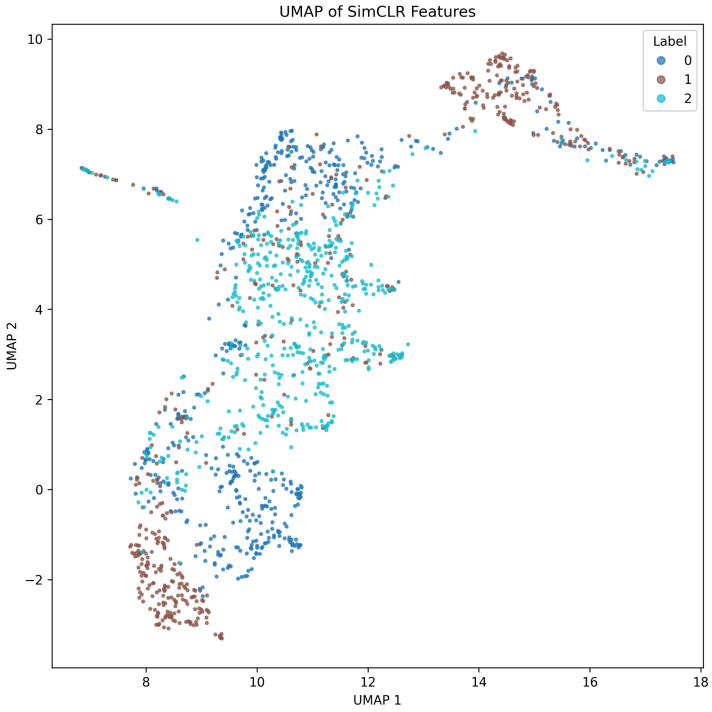
Table 1: DCGAN training results by image grade.

SimCLR

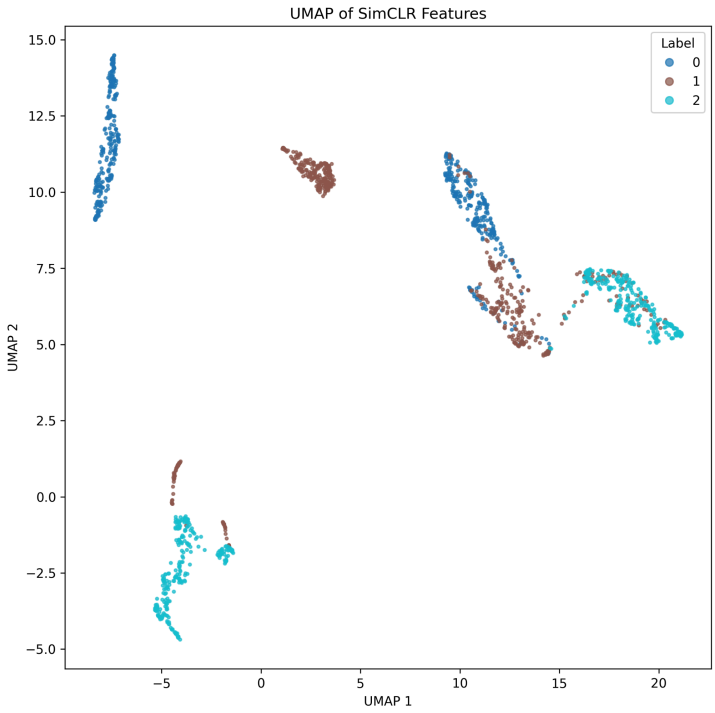
I used UMAP to help evaluate my SimCLR results by visualizing the learned feature representations in 2D. In Figure 4a, you can see the separation of classes using the regular proposed transforms of SimCLR. The points from different classes are scattered throughout the space, with significant overlap between the normal, low-grade, and high-grade groups. This blending suggests that the feature representations fail to separate the tumor grades in a meaningful way.

In contrast, in Figure 4b, the UMAP plot shows a much clearer separation of classes. This figure was created after modifying the transforms to better suit medical imaging data. By removing transformations that distorted the anatomical features of the images, the SimCLR model was able to produce features that better separated the grades of tumors. This confirms that the choice of augmentations is critical in self-supervised learning, especially when working with medical images.

The effect that SimCLR had on the downstream tasks will be discussed later in the supervised learning section.



(a)



(b)

Figure 4: (a) UMAP of SimCLR features before applying domain-appropriate transformations. (b) UMAP after adjusting transformations for medical imaging. Class 0 refers to normal images, 1 refers to low-grade, and 2 refers to high-grade.

Sensitivity Analysis

To test how sensitive the DCGAN model is to different hyperparameters, I tested different values for the learning rate and number of epochs, as well as set bias to True for the Conv2D layers. Tests were done using the normal image data. All results can be seen below in Table 2, where the changed hyperparameters are highlighted in yellow.

Learning Rate	Epochs	Bias	Best LPIPS Score	Epoch of Best Score
0.0002	300	False	0.1123	290
0.0001	300	False	0.1152	210
0.0003	300	False	0.1147	270
0.0002	150	False	0.1384	150
0.0002	400	False	0.1078	180
0.0002	300	True	0.1114	280

Table 2: Results of the DCGAN sensitivity analysis. Changed hyperparameters are highlighted in yellow for clarity.

Reducing the learning rate by 50% or increasing it by 50% did not have a large difference on the LPIPS score.

Cutting the number of epochs in half, however, did have an impact, and due to never reaching as low of a score, likely means that the model was undertrained. Increasing the epochs to 400 improved the LPIPS score, but that best score was reached at 180 epochs, meaning that longer training did not help further. Lastly, changing from bias = False to bias = True for the Conv2D layers had minimal effect on the best score. Overall, it can be concluded that out of the hyperparameters tested, the DCGAN model is only sensitive to the number of epochs.

Supervised Learning

Methods

Three different deep learning image classifications models were created and used for the supervised learning portion of this project. The first choice was CNN, as it has been the go-to for image classification for many years, and particularly in the field of medical imaging. Its use of convolutional layers enables the model to detect local features like edges, textures, and shapes, which is especially important for tasks like tumor detection. The next choice was MLP-Mixer, which has become popular in recent years. It forgoes the use of convolutions or attention for two MLP layers. One mixes features within each individual image patch, while the other mixes information across different patches to capture spatial relationships in the image. This allows it to achieve competitive scores on image classification tasks (Tolstikhin et al., 2021). Lastly, I chose ViT, which has also recently gained popularity. This method utilizes the power of transformers, which allows it to achieve strong performance compared to convolutional networks, while requiring substantially fewer computational resources to train (Dosovitskiy et al., 2020). For all methods, pre-trained models were used due to the small size of the dataset.

To optimize model performance, I conducted hyperparameter tuning through a grid search and finetuned the models by tracking validation accuration across different epochs. For the grid search, I tried different values for the learning rate and weight decay to see which would produce the best model. The best results for all three models were achieved using a weight decay of 0.01. The optimal learning rate was 0.003 for both the MLP-Mixer and the ViT, while the CNN performed best with a learning rate of 0.001. The learning rate and weight decay values that were tested are shown in Table 3. I also chose to freeze the encoder during training, as every model consistently returned better results with this option. For the number of epochs, I started with 20 epochs across all models. As can be seen in Figure 5, the CNN validation accuracy dips after 10 epochs and then starts to go back up after 15. Given this dip and delayed improvement, I opted to stop training at 10 epochs to prevent overfitting and reduce unnecessary training time. The MLP-Mixer validation accuracy peaks at 10 epochs and the ViT validation accuracy has already reached its peak at 5 epochs, therefore these models were stopped at 10 and 5 epochs, respectively, to capture the peak performance and avoid overfitting.

Model	Learning Rates	Weight Decay Values	Best Learning Rate	Best Weight Decay
CNN	[0.001, 0.003, 0.01]	[0.01, 0.001, 0.0001]	0.001	0.01
MLP-Mixer	[0.001, 0.003]	[0.01, 0.05, 0.1]	0.003	0.01
ViT	[0.001, 0.003]	[0.01, 0.05, 0.1]	0.003	0.01

Table 3: Learning rate and weight decay values tested during hyperparameter tuning for each model, along with the best-performing combination for CNN, MLP-Mixer, and ViT. The hyperparameter options were selected based on values reported in the original papers for each model architecture (Radford et al., 2015; Tolstikhin et al., 2021; Dosovitskiy et al., 2020).

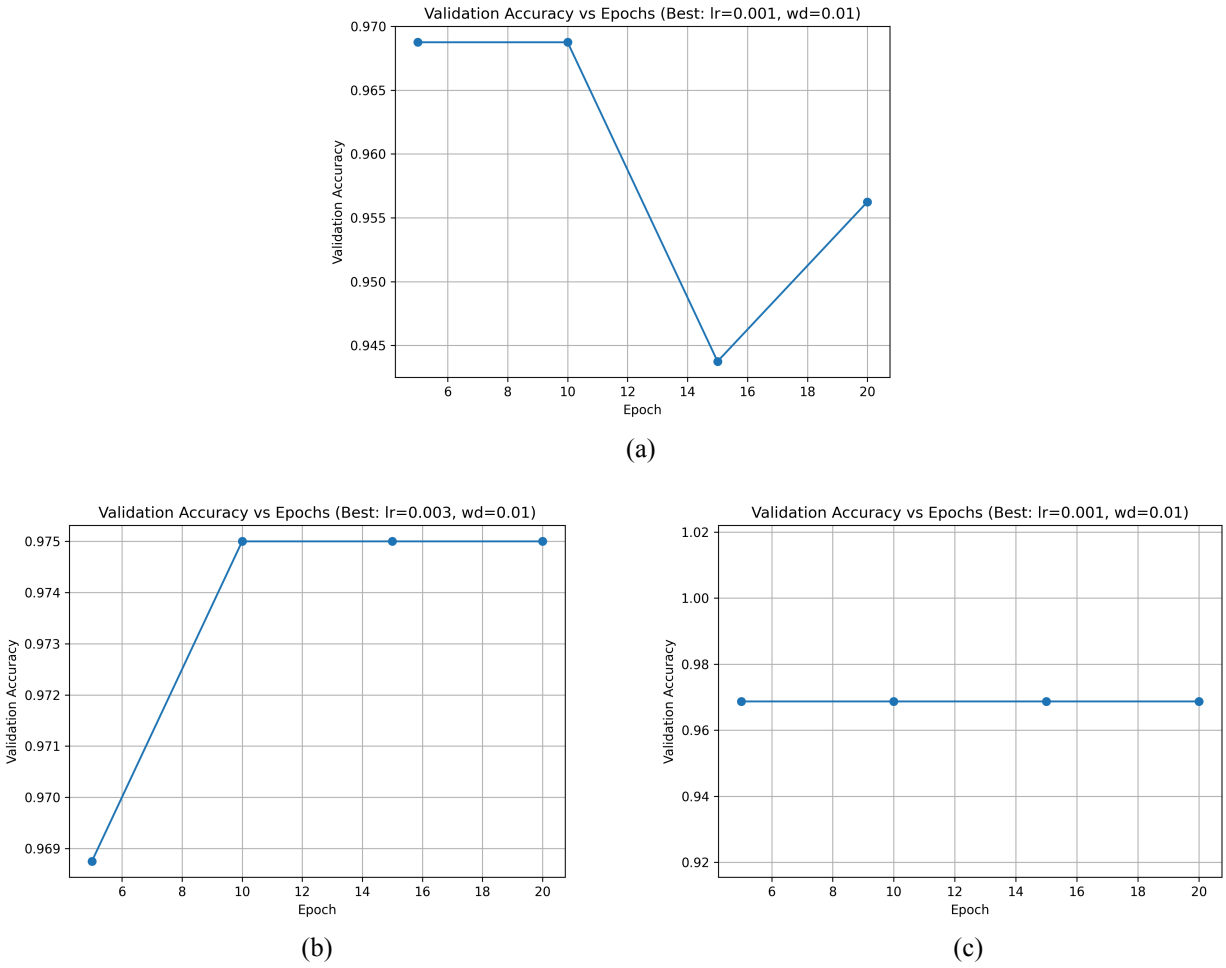


Figure 5: Validation accuracy across epochs for (a) CNN, (b) MLP-Mixer, and (c) ViT during hyperparameter tuning.

To further check for overfitting, I monitored both validation and test accuracy throughout training. Test accuracy remained close to validation accuracy for all models, with only a slight dip, suggesting that overfitting was not a significant concern. The CNN reached a validation accuracy of 96.88% and a test accuracy of 96.25%, the MLP-Mixer reached 97.5% and 95%, and the ViT reached 96.88% and 95.63%.

Evaluation

In addition to accuracy, model performance was evaluated using precision, recall, F1-score, and confusion matrices. Accuracy was chosen because it provides an overall measure of how often the model predicts correctly, but it can be misleading on its own. This is why it was supplemented with precision, recall, and F1-score. Precision measures how many of the images that were predicted to be a particular class were actually correct, and in the case of predicting tumors, a false positive could mean that expensive tests are done that were not needed. More importantly though, recall will capture whether a model catches all the true positives. Put simply, it indicates whether tumors have been correctly identified. Missing a tumor would have serious consequences, so a high recall score is especially important and even more so than the cost or burden of follow-up tests that might result from a false positive. The F1-score balances the tradeoffs between precision and recall and provides a single, interpretable metric of class-level performance. For F1 scores, macro averages were calculated to reflect equal importance across classes. Finally, confusion matrices offer a detailed view of the precision and recall metrics and can help identify exactly where the model is misclassifying inputs.

CNN Results

Metric	Normal (Class 0)	Low-grade (Class 1)	High-grade (Class 2)	Metric	Score
Precision	1.0	0.9800	0.9123	Accuracy	0.9688
Recall	1.0	0.9074	0.9811	Macro F1-Score	0.9687

Table 4: Test set performance metrics for the CNN model.

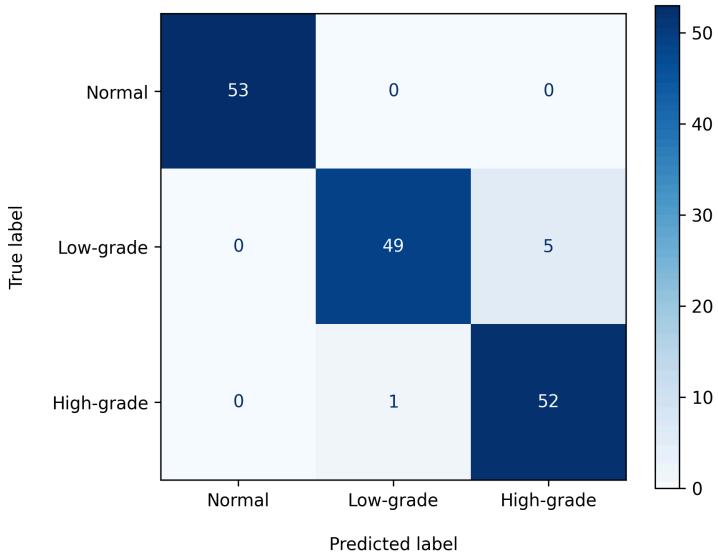


Figure 6: Confusion matrix for CNN model on the test set.

MLP-Mixer Results

Metric	Normal (Class 0)	Low-grade (Class 1)	High-grade (Class 2)	Metric	Score
Precision	1.0	0.9423	0.9091	Accuracy	0.9500
Recall	1.0	0.9074	0.9434	Macro F1-Score	0.9502

Table 5: Test set performance metrics for the MLP-Mixer model.

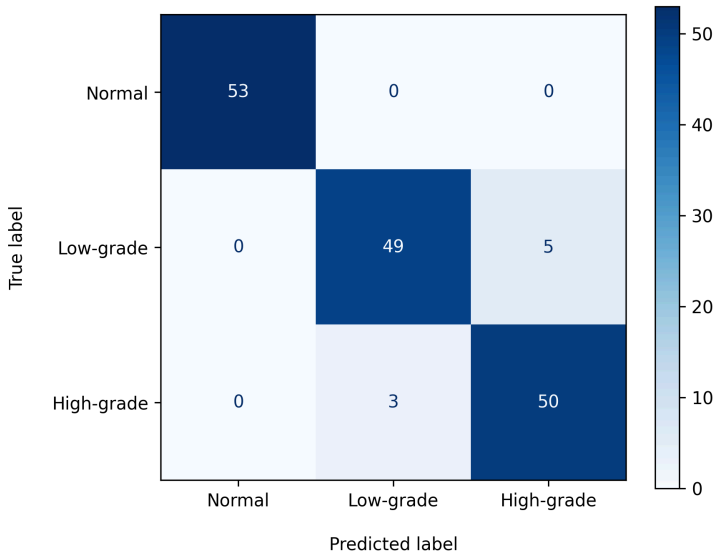


Figure 7: Confusion matrix for MLP-Mixer model on the test set.

ViT Results

Metric	Normal (Class 0)	Low-grade (Class 1)	High-grade (Class 2)	Metric	Score
Precision	0.9815	0.9796	0.9123	Accuracy	0.9563
Recall	1.0	0.8889	0.9811	Macro F1-Score	0.9560

Table 6: Test set performance metrics for the ViT model.

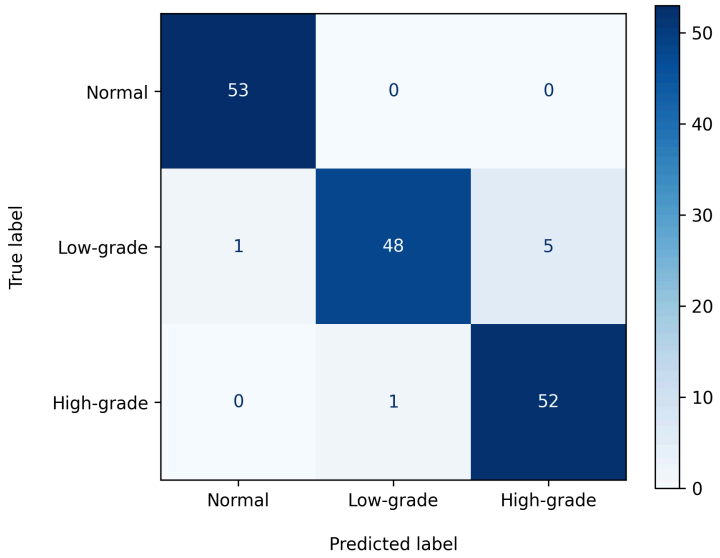


Figure 8: Confusion matrix for ViT model on the test set.

With an accuracy of 96.88%, the CNN model performed the best out of the three, with the MLP-Mixer and ViT close behind at 95% and 95.63%, respectively. These numbers are so close, however, that it's important to look at further metrics to assess which is best. Precision is considered, but given that all precision metrics are higher than 90%, making recall the more critical metric for ensuring early tumor detection. In terms of recall performance, the, ViT lagged slightly behind the MLP-Mixer for low-grade images, whereas MLP-Mixer underperformed by close to 4% on high-grade images. Knowing what images were misclassified as can help us determine which tradeoff is worth it. As can be seen in Figures 7 and 8, MLP-Mixer only misclassified low-grade images as high-grade, whereas ViT misclassified one low-grade image as normal. Failing to detect a tumor would have much more dire consequences than getting the grade of it incorrect. Therefore, in the context of tumor classification, the MLP-Mixer outperforms ViT. That puts the rankings in order from best to worst as CNN, MLP-Mixer, and ViT.

Ablation Analysis

To get insight into which features of my model were contributing to prediction success, I ran two ablation tests. The first was removing all the generated images from the DCGAN and the second was removing the SimCLR encoder and replacing it with a standard pretrained encoder. When the GAN images were removed, accuracy dropped over 6% and recall for low-grade images dropped close to 14%. The recall for high-grade images also dropped close to 6%. This indicates that the generated images were quite valuable, especially in helping the low-grade images. Removing the SimCLR encoder also led to reduced accuracy and recall, though less significant. There was about a 1% drop in accuracy and only the recall for low-grade images was affected, with a drop of about 4%. Together, these results show that both the GAN and the SimCLR encoder were important to the success of the classifier.

Remove	Accuracy	Recall		
		Normal	Low	High
Nothing	0.9688	1.0	0.9074	0.9811
GAN Images	0.9000	1.0	0.7692	0.9259
SimCLR Encoder	0.9562	1.0	0.8696	0.9811

Table 7: Ablation analysis results for the CNN model.

Sensitivity Analysis

For the sensitivity analysis, I focused on the best-performing model of the three, which was the CNN. I tested different values for learning rate, weight decay, and number of training epochs to evaluate whether model performance was sensitive to these hyperparameters. The different tests can be seen in Table 8 with parameter changes highlighted in yellow for clarity. I evaluated performance using only accuracy and recall, as those are the most important metrics in this context.

The results revealed that the model's performance remained consistent across all combinations. Validation accuracy plateaued at 96.88% regardless of the chosen hyperparameters and recall values for all classes remained unchanged. This stability can be attributed to a few key reasons. The encoder was already pretrained with SimCLR and frozen during training, meaning the classifier had limited capacity and adjusted quickly. The dataset was relatively small, and the classifier was a simple linear layer, which meant the model reached its optimal performance early. These results suggest that the model was not particularly sensitive at all to these hyperparameters.

Learning Rate	Weight Decay	Epochs	Accuracy	Recall		
				Normal	Low	High
0.001	0.01	10	0.9688	1.0	0.9800	0.9123
0.01	0.01	10	0.9688	1.0	0.9800	0.9123
0.003	0.01	10	0.9688	1.0	0.9800	0.9123
0.001	0.001	10	0.9688	1.0	0.9800	0.9123
0.001	0.0001	10	0.9688	1.0	0.9800	0.9123
0.001	0.01	15	0.9688	1.0	0.9800	0.9123
0.001	0.01	20	0.9688	1.0	0.9800	0.9123

Table 8: Sensitivity analysis results for the CNN model. Changed hyperparameters are highlighted in yellow for clarity.

Failure analysis

To better understand model performance, I analyzed confusion matrices and recall metrics across class, and used Local Interpretable Model-agnostic Explanations (LIME) to help explain patterns in misclassifications. As can be seen in the confusion matrices in Figures 6, 7, and 8, and is confirmed in the recall metrics in Tables 4, 5, and 6, recall was excellent overall except for low-grade images. Looking at just the CNN model, as shown in the confusion matrix in Figure 6, there were five low-grade images that were misclassified as high-grade and one high-grade image that was misclassified as low grade.

I quickly realized that the most likely reason for misclassification of the low-grade images is that they all came from the smallest data, the UCSF data. This is significant because although I had a balanced dataset in terms of the different grades of tumors, the low-grade data came from two different sources and the UCSF data only provided 56 images. Another 56 generated images were created from those, totaling 112 similar images. These accounted for around 7% of the images, whereas the other sources accounted for around 26% (the other low-grade images) or 33% (normal and high-grade images). With so few real examples to learn from, the model lacked the diversity and quantity of data needed to generalize well to this subset. This imbalance likely contributed to the low-grade images, and specifically the low-grade images from that source, being misclassified the most and at a much higher rate than other grades. Removing these images from the original data would likely fix the misclassifications with the low-grade images.

The reason why the high-grade image was misclassified was not as clear, so I turned to LIME to try to find one. It works by highlighting regions of the image that contributed most to the prediction. In Figure 9, the pixels in green with yellow borders indicate those areas. However, since we do not know where the tumor is in the original image, it's difficult to determine whether these highlighted areas are focused on the actual tumor or irrelevant regions. This limited my ability to confidently interpret the high-grade image, but if these values were available in the future, they would be a great help in determining the true source of the misclassifications.

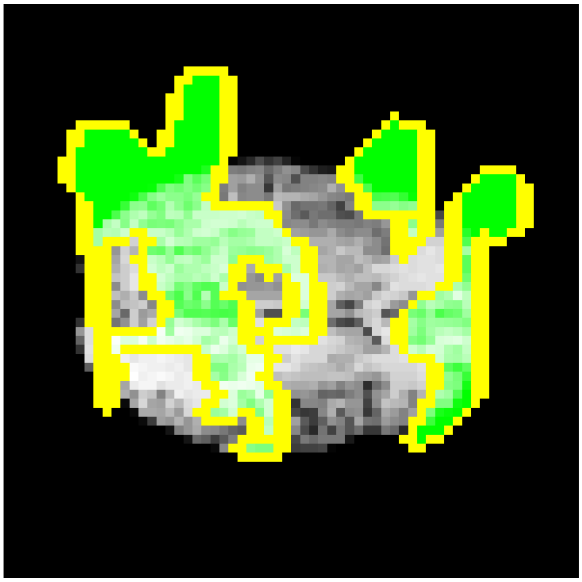


Figure 9. LIME explanation for a misclassified high-grade glioma image using the CNN model. The true label was high-grade (2), but the model predicted low-grade (1).

Discussion

Unsupervised

Through the unsupervised learning part of this project, I gained a deeper understanding of what tasks to use for smaller datasets and how to implement them. I had learned about GANs previously, but had never implemented one from scratch. It ended up being a lot more work than I originally thought it would be. I did not realize how much hyperparameter tuning and general tweaking would be involved and expected to get great results from the get-go. That was certainly not the case, and when my FID scores were initially so poor, I had to pivot quickly. I had to continuously research more and more into how to get the best outcome. Finding out that FID was not always the best metric was surprising to me, as that is what I have heard used with GAN models.

SimCLR was completely new to me, so using it taught me a brand new method for improving supervised classification tasks. It helped me appreciate the importance of feature representation and the effect that unsupervised pretraining can have on downstream supervised learning. I learned how careful design of transformations is essential and needs to always make sense within the context of your data. Unlike the GAN, the results were strong from the start, which surprised me. It took some time to set everything up, but once it was working, my UMAP visualization showed how well-separated the different classes were, confirming that the SimCLR features captured meaningful representations in the data.

Given more time and resources, I would train my DCGAN using a larger dataset to improve my outputs and help with accuracy downstream. The original plan was to use two large datasets from a well-known cancer imaging repository, but unfortunately this was not possible due to accessibility issues and the datasets had to be changed last minute. Having more data would improve my results, but if I were not able to get larger datasets, I would still focus on improving the quality of my DCGAN outputs by reducing the LPIPS score as much as possible. That way I could produce more generated images, without the concern that they are not close enough to the real ones, and therefore still have a large dataset.

Supervised

The supervised learning portion of this project taught me how to carefully tune different classification models and how to compare the end results. This was my first time implementing a train, validation, test split and utilizing the

validation set to tune hyperparameters and monitor for overfitting. The visualizations of validation accuracy versus epochs allowed me to identify the optimal number of epochs to achieve strong performance while avoiding overfitting. Comparing the different metrics, such as accuracy and recall, taught me the importance of weighing tradeoffs for your specific data. The numbers must be interpreted within the confines of your specific context in order to give them meaning and significance.

I was surprised by how close the results were across all three models and that the results were immediately very strong. Given the differences in architecture and the small size of the data, I expected there to be more variation. My original hypothesis was that CNN would perform the best and ViT the worst. I also thought that there would be more of a spread in the results, but the accuracy results were all within 2% of each other. Nearly all metrics were above 90% from the first run. A large portion of this is likely due to using pretrained models, but as seen in the ablation test, the generated images and SimCLR encoder played a large role too.

Overall, the supervised learning portion was much more straightforward compared to the unsupervised part, but I did run into a challenge with memory constraints. Due to the high-resolution of the medical images I was working with, loading them all into memory at once was not feasible. When I created my custom dataset class, I needed to implement lazy loading to address this issue. This allowed me to load each image as needed during training, which significantly reduced memory usage and made it possible to train my models on a standard GPU without crashing.

Similarly to the unsupervised learning portion, if I had more time and resources, I would train the classifiers using a larger and more consistent dataset. One of the low-grade sources only provided 56 images, which was a very small number for the model to learn from, even after augmentation. As a result, the model may not have been exposed to enough variation within that dataset to learn distinguishing features. This likely contributed to the model's difficulty in correctly classifying low-grade images from that source.

Ethical Considerations

There are always important ethical considerations when working with medical data. One of the most prominent concerns is around the use of personally-identifiable information (PPI). Thankfully, none of the data used in this project has any PII, but that does not mean that there are not other ethical issues to consider. The information obtained at the time of collection is only part of the concern for this project**.** Making sure that the model is fair and how the model will be used after it is produced are also extremely important ethical considerations and were considered during the unsupervised and supervised learning parts of the project, respectively.

In the unsupervised learning portion of this project, a DCGAN was built and used to create additional MRI images to augment the dataset. The main ethical concern here would be the model possibly reinforcing any biases within the data. Steps were taken to evaluate potential sources of bias as much as possible. Class imbalance was addressed by creating a dataset with equal numbers of images for each class. Consistency throughout MRI modality was maintained as best as possible by only using two types of images- FLAIR and T1-weighted MRIs. Different datasets both using only T1-weighted images were originally going to be used for this project, but were unavailable due to a technical issue with the hosting website. No demographic data about age, sex, race, etc. was collected, however, which limits the ability to assess fairness across subpopulations. This could be addressed in a different project by using a dataset that provides this information.

The SimCLR method was also used, which relies heavily on data augmentations to create different views of the same image. The original SimCLR paper uses augmentations such as cropping, flipping, and color jitter that would not be valid for medical imaging. These issues were addressed by removing flipping and color jitter from the transformations and performing only a minor cropping.

For the supervised learning portion of the project, CNN, MLP-Mixer, and ViT classifiers were used to predict the presence of a glioma and, if present, its grade. It is important to note that the models created are not a diagnostic system and should not be used in clinical settings without proper validation. The data that the model was trained on was limited and the model may not generalize well to other imaging modalities or real-world applications outside this controlled dataset.

References

ApX Machine Learning. (n.d.). *Interpreting FID scores* [Web tutorial]. Retrieved June 6, 2025, from https://apxml.com/courses/generative-adversarial-networks-gans/chapter-5-evaluation-of-gans/interpreting-fid-scores

Benz, P., Ham, S., Zhang, C., Karjauv, A., & Kweon, I. S. (2021). *Adversarial robustness comparison of Vision Transformer and MLP-Mixer to CNNs*. arXiv preprint arXiv:2110.02797. https://arxiv.org/abs/2110.02797

Brain Tumour Research. (n.d.). *Glioma: Types of brain tumours*. Retrieved June 20, 2025, from https://braintumourresearch.org/pages/types-of-brain-tumours-glioma?srsltid=AfmBOooy20A3S3FLiIT_itHrtn49vjIMQtWp1KltTyRg4K4RfqGgqei-

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, July 1). *A simple framework for contrastive learning of visual representations* [Preprint]. arXiv. Retrieved June 6, 2025, from https://arxiv.org/abs/2002.05709

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020, October 22). *An image is worth 16×16 words: Transformers for image recognition at scale* [Preprint]. arXiv. Retrieved June 6, 2025, from https://arxiv.org/pdf/2010.11929

Goceri, E., Yılmaz, G., Uysal, İ., Ergen, M., & Doğan, Z. (2024). *An efficient network with CNN and transformer blocks for glioma grading and brain tumor classification from MRIs*. Expert Systems with Applications. Advance online publication. https://doi.org/10.1016/j.eswa.2024.119518

Information eXtraction from Images (IXI) dataset. (n.d.). *IXI dataset* (RRID: SCR_005839) [Data set]. Retrieved June 26, 2025, from http://brain-development.org/ixi-dataset/

Lightning AI. (n.d.). *Learned perceptual image patch similarity (LPIPS)* [Documentation]. Retrieved June 6, 2025, from https://lightning.ai/docs/torchmetrics/stable/image/learned_perceptual_image_patch_similarity.html

Medicus Healthcare Solutions. (2025, February 17). *The radiologist shortage: Addressing the gap between supply and demand*. Retrieved June 20, 2025, from https://medicushcs.com/resources/the-radiologist-shortage-addressing-the-gap-between-supply-and-demand

Parisot, S., Darlix, A., Baumann, C., Zouaoui, S., Yordanova, Y., Blonski, M., … Chemouny, S. (2015, November 19). *Diffuse Low-grade Glioma Database* [Data set]. Figshare. http://dx.doi.org/10.6084/m9.figshare.1550871

PyTorch. (n.d.). *DCGAN faces tutorial* [Documentation]. Retrieved June 6, 2025, from https://docs.pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised representation learning with deep convolutional generative adversarial networks* [Preprint]. arXiv. Retrieved May 28, 2025, from https://arxiv.org/pdf/1511.06434

Skandarani, Y., Jodoin, P.–M., & Lalande, A. (2023, March 16). *GANs for medical image synthesis: An empirical study* [Research article]. *Journal of Imaging*, 9(3), 69. Retrieved June 6, 2025, from https://pmc.ncbi.nlm.nih.gov/articles/PMC10055771/

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., & Dosovitskiy, A. (2021, June 11). *MLP-Mixer: An all-MLP architecture for vision* [Preprint]. arXiv. Retrieved June 6, 2025, from https://arxiv.org/pdf/2105.01601

UCSF Center for Intelligent Imaging. (2022, December 13). *University of California San Francisco preoperative diffuse glioma MRI (UCSF-PDGM) dataset* [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/tcia.bdgf-8v37

Wang, Z., Li, T., Zheng, J.-Q., & Huang, B. (2022). *When CNN meet with ViT: Towards semi-supervised learning for multi-class medical image semantic segmentation* [arXiv preprint arXiv:2208.06449]. https://doi.org/10.48550/arXiv.2208.06449

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018, April 10). *The unreasonable effectiveness of deep features as a perceptual metric* [Preprint]. arXiv. Retrieved June 6, 2025, from https://arxiv.org/pdf/1801.03924