# BRAIN TUMOR DETECTION AND CLASSIFICATION FROM MRI IMAGES

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Anjaneya Teja Sarma Kalvakolanu

March 2021

COMMITTEE MEMBERSHIP

TITLE: Brain Tumor Detection and Classification from MRI images

AUTHOR: Anjaneya Teja Sarma Kalvakolanu

DATE SUBMITTED: March 2021

COMMITTEE CHAIR: Franz Kurfess, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Maria Pantoja, Ph.D.
Associate Professor of Computer Science

COMMITTEE MEMBER: Theresa Migler, Ph.D.
Assistant Professor of Computer Science

ABSTRACT

Brain Tumor Detection and Classification from MRI images

Anjaneya Teja Sarma Kalvakolanu

A brain tumor is detected and classified by biopsy that is conducted after the brain surgery. Advancement in technology and machine learning techniques could help radiologists in the diagnosis of tumors without any invasive measures. We utilized a deep learning-based approach to detect and classify the tumor into Meningioma, Glioma, Pituitary tumors. We used registration and segmentation-based skull stripping mechanism to remove the skull from the MRI images and the grab cut method to verify whether the skull stripped MRI masks retained the features of the tumor for accurate classification. In this research, we proposed a transfer learning based approach in conjunction with discriminative learning rates to perform the classification of brain tumors. The data set used is a 3064 T MRI images dataset that contains T1 flair MRI images. We achieved a classification accuracy of 98.83%, 96.26%, and 95.18% for training, validation, and test sets and an F1 score of 0.96 on the T1 Flair MRI dataset.

# ACKNOWLEDGMENTS

Thanks to:

- Dr. Franz Kurfess for advising me and helping me through out my thesis.
- My family for supporting me through out my grad school.
- Cal Poly for providing me resources to complete my thesis.
- Figure Share inc for helping me with the data set.
- Andrew Guenther for creating the latex template.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# Chapter 1

# INTRODUCTION

The number of persons infected by brain tumors increases every year. Tumors are caused by abnormal growth of cells. Brain tumors can be benign (noncancerous tumors) or malignant (cancerous). They are also classified as primary and secondary. Primary tumors start in the brain or Central Nervous Systems whereas the secondary tumors spread from other body parts into the brain. Depending on the degree of abnormality of brain tissue, the tumors are typecast into four grading levels. Tumors with 1 and 2 are low grades which are less dangerous. 3 and 4 grade tumors are high-grade tumors that are highly susceptible to cancer.

Primary tumors have several types amongst 36.1% all primary tumors are referred to as meningioma that is found near the top and outer curve of the brain. Meningioma is a slowly growing non-cancerous tumor that causes seizures and visual problems. Glioma is an abnormal growth in glial cells present around the neurons in the brain. Pituitary tumors grow in pituitary glands that affect body functions. Meningioma is iso-dense dural-based masses developed at the meninges of the three layers of protecting tissue of the brain and spinal cord, whose diagnosis depends on its anatomical location, shape, and appearance of cells. Pituitary tumors are abnormal mass growth in cells around the surface of the pituitary gland located at the base of the skull.

Early detection of primary brain tumors is highly complex because of their position, shape, and diversity. Non-invasive imaging techniques are based on the absorption properties of tissues. The delineation of absorption rates is very important for the precise imaging of tumors. The three types of tumors concentrated on this case study

are not much discriminative in terms of their absorption levels (Pixel gray level). Hence the shape of the tumors will play a major role in classifying them.

Deep learning has evolved in recent years and is very successful in problems that involve pattern recognition or shape detection. Transfer learning made its way into Deep Learning which reduced the hectic work of training the models from scratch. For this reason, we opted for transfer learning with Resnet 50 that is pre-trained with an Image Net database that consists of more than 1000 different target classes. To make it suitable for brain tumor detection we utilized Discriminative Learning rates with Grab Cut and Skull stripping in pre-processing that allows the final layers to be only trained based on the MRI data while the basic pattern recognition is retained from the Resnet50 by using pre-trained weights.

### 1.1 Brain Tumor

The brain is a vital organ in the human body and responsible for control and decision making. As the managing center of nervous systems, this part is very essential to be protected from any harm and illness. Tumors are the predominant infections caused by abnormal growth of cells that damages the Brain. Meningioma, Glioma, and Pituitary are brain tumors as opposed to the other types. Meningiomas are mostly a non-cancerous class of tumors that often develop in the narrow walls that usually surround the brain [22]. Brain tumors are one of the life-threatening diseases that can directly affect human lives. The correct understanding of brain tumor stages is an important task for the prevention and cure of illness. To do so, Magnetic Resonance Imaging (MRI) is widely used by radiologists to analyze brain tumors. The result of this analysis reveals whether the brain is normal or abnormal. On the other hand, it identifies the type of tumor in the case of abnormality[35]. With the advent of machine

learning, the processing of MR images to have a for fast and accurate detection of brain tumors matter.

## 1.2 Machine Learning and Feature Extraction

In the beginning, approaches consisted of three steps: pre-processing of MR images, , feature generation, and extraction and classification. The median filter was used for the improvement of image quality and preserving the edges in the pre-processing stage [37]. Segmentation of images with clustering methods such as k-means, fuzzy C-means, etc. generates beneficial features from images. Image segmentation plays an essential role in analyzing and interpreting images. It has vast applications in brain imaging, such as tissue classification, tumor location, tumor volume estimation, blood cell delineation, surgical planning, matching. In [17], a brain tumor segmentation is applied by using a CNN to 3D MR images. Automatic detection of the anatomical structure of the brain with a deep neural network is proposed in [33]. In a voting technique for an ensemble of visual appearances such as intensity and adaptive shape modes using a combination of discrete Gaussian and higher-order patterns such as Markov-Gibbs, random field classification is used. A hybrid deep auto-encoder with Bayesian fuzzy clustering-based segmentation approach is developed in [32]. In this work, after denoising with a non-local mean filter, a Bayesian fuzzy clustering approach is utilized for the segmentation of brain tumors. In [11] the 2D MRI mages are partitioned into the left and right hemisphere and statistical features such as mean, homogeneity, absolute value, and inertia are computed for the Support Vector Machine (SVM) classifier. Due to the vast number of features in step two, most studies consist of a further step to extract features with more valuable information using algorithms like principal component analysis (PCA) or SIFT detectors and SURF descriptors [18]. In [10] after a hybrid feature extraction with a covariance matrix, a

regularized extreme learning is used to classify brain abnormality. Evolutionary algorithms like particle swarm optimization (PSO) also used in [30] for selecting from a fusion of features. Classification method such as k-nearest neighbors, decision trees, Support Vector Machine (SVM), the Naive Bayes, expectation-maximization, and the random forest is the popular machine learning technique on image analysis [38]. In [14] feature extraction is used for hybrid Functional Near-Infra Red Spectroscopy (fNIRS) and Electro Encephalo Graphy (EEG) brain-computer interface and classified with SVM and Linear Discriminant Analysis (LDA). Nowadays, Convolutional Neural Networks (CNN) become popular in terms of feature extraction in different studies such as medical images, video analysis, and natural language processing. The main aspect of CNN is the ability to recognize the foremost patterns and information from the training images. For instance, VGGNet[31], GoogleNet [36], and AlexNet [19] are successful architectures in image classification that widely used for medical images like brain abnormality detection. In [4] pre-processing and data-preparation with 3D filters and CNNs with multi-path and cascade architectures are studied. A pixel CNN architecture is applied to produce a variety of new portraits of a person with different expressions and poses. In [39] cascade CNNs are iteratively generating a room decoration from scratch. Since CNN has a high computational cost, researchers try to discover new computationally simple models with accurate tumor classification.A typical approach is to use an ensemble of small collaborative learners instead of a complex network, to meet fast training execution and convergence. The learning process of these peer networks can be independent of each other or have a mutual dependency. One of the popular challenges in machine learning task is to estimate the distribution of data. For example, there are hard-coded correlations among image pixel and its neighbors, where is difficult to find without prior information. The auto-regressive models are data-driven estimators that find such correlations with general data. The generated output of these models is improved images conditioned

on noisy or incomplete data. An admissible density estimator likely solves a variety of classification, regression, missing data, and similar problems.

## 1.3 Paper Organization

This paper is organized as follows, Background is introduced in Chapter 2 with an in-depth explanation of techniques used in this thesis, followed by related work which includes the work that is related to this thesis described in Chapter 3, proposed methodology and system diagram is discussed in Chapter 4. Experimental Setup is described in Chapter 5 followed by results in Chapter 6, Chapter 8 contains future work, and Chapter 9 is used for the conclusion.

Chapter 2

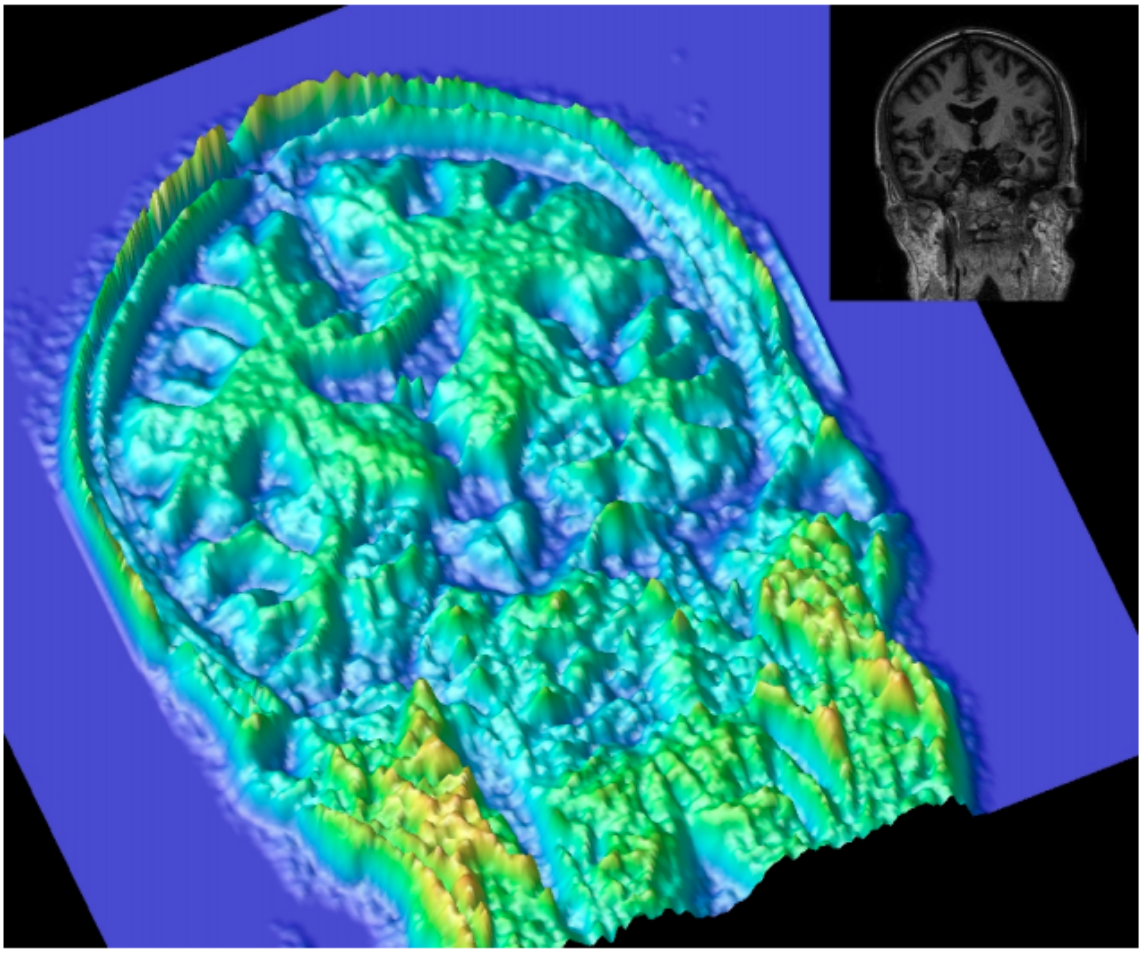
# BACKGROUND

## 2.1 Registration and Segmentation for Skull Stripping

Extraction of the cerebrum, cerebellum, and brain stem from MR images is an important initial step in neuroimaging. We used an algorithm [7] that solve this difficult problem, often referred to as skull stripping, which is novel for its use of registration, segmentation, and morphological operations. In most previous approaches, the goal of skull stripping was to segment out Grey Matter(GM) and White Matter(WM) from all other tissues, without any concern for an accurate representation of the GM boundary with the Cerebrospinal Fluid (CSF) and other non-WM tissue. As such, we wish to remove the skull, meninges, and other tissues without excising any cortical surface GM or WM. The CSF surrounding the cortical surface is important to define the GM boundary, and the method we used consistently includes a thin layer of CSF around the brain.

The Grey Matter and White Matter we are interested in are connected by the scheme that there is a path connecting any two GM/WM Voxels such that from the point of highest intensity along the path to either endpoint the intensity is never increasing. Voxels are considered to be connected if they are on the same peak [7].

This can be understood by taking a slice of an MRI dataset and using the intensities as height information. With this representation, Voxles are connected if they are on the same hill and disconnected if they are separated by a valley, see 2.1 for an illustration.



**The heights and color schemes are derived from the image intensities [7]**

**Figure 2.1: Voxels mapping vs intensity**

With this concept it becomes apparent that if we can reasonably isolate the WM (hill peaks) we can perform a morphological dilation based on descending this hill to the valley represented by the CSF and image background. We will also add an appropriate stopping criterion in those instances when we are traversing a plane, with respect to image intensities.

As the first step, to identify the crest of the hill corresponding to the GM/WM connected component an ABA registration [26] between the subject brain and the atlases are performed. Here, an atlas is the combination of an intensity image and its segmented image that the algorithm uses for registration. ABA registration maximizes mutual information with an underlying deformation field modeled on radially symmetric basis functions.

A mask for each atlas is then transformed into the subject space based on the output of the registration. The mask is a representation of the cerebral cortex, cerebellum, and brain stem. We then generate a probability mask given in the subject space at voxel $i$ as,

$Pi =$ (No of Masks that include $i$)$/N$, see Fig 2.2(a).

In addition, a tissue segmentation of the whole head MRI using FANTASM [25] is generated. It is a tissue classification algorithm that is an extension of the adaptive fuzzy c-means algorithm. We use four classes within FANTASM, two of these classes ($\lambda_2$ and $\lambda_3$) are an approximate representation of WM and GM. other tissue classes break down into a class made up of CSF, bone, and background ($\lambda_1$) and another ($\lambda_4$) dominated by the skin and adipose tissue and a small number of WM voxels.

Fig. 2.2(b) shows all four tissue classes for a particular dataset. The algorithm then combines the registration and segmentation in order to have the best of what either

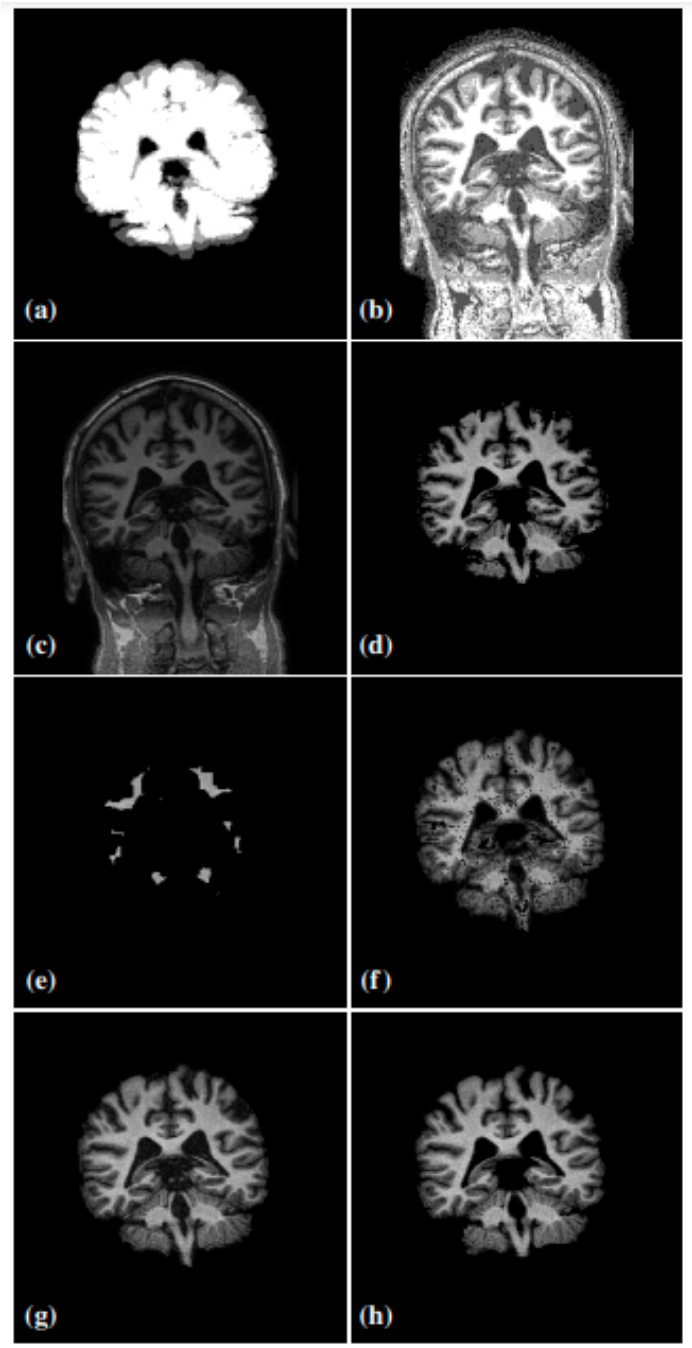



**Figure 2.2: Steps of skull stripping**
**For a given data set we show (a) the probability mask (b) the tissue classification, (c) the original image, (d) the initial mask, (e) the mask after erosion and retaining the largest connected component, (f) the mask prior to the modified morphological closing, (g) Mask obtained and (h) Original Mask. [7]**

has to offer. So the initial mask M is produced using the 2.1 criterion based on both the registration and segmentation:

$$\{i|i \in \Gamma_3, P_i \geq \frac{N-1}{N}\} \cup \{i|i \in \Gamma_2, P_i = 1\} \tag{2.1}$$

This initial segmentation provides a representation of the WM and subcortical GM with a high degree of confidence 2.2(c). There are two problems with this initial mask: tissue from the neck and around the base of the brain stem included in the mask, secondly we can have gross errors in the registration that lead to portions of dura and skull being included. To deal with these problems, An aggressive morphological erosion operation on the current mask with a structuring element of 3x3x3 voxels is performed. This erosion is performed twice and the largest 6-connected component is found. That component remains in the mask and excises all other connected components from the mask, shown in Fig. 2.2(e).

The next major step is to perform our hill decent in a reasonable manner, using both the registration and segmentation. Starting from the eroded initial mask, we add voxel $i$ on the boundary to our mask $M$ if it meets either of the two following criteria 2.2 and 2.3:

$$i \in \Gamma_2 \cup \Gamma_3, I_j \geq I_i, P_i \geq \frac{1}{N} \quad for j \in M \cup \Gamma_2 \cup \Gamma_3 and i \in N_{18}(j) \tag{2.2}$$

$$i \in \Gamma_1 \cup \Gamma_2 \cup \Gamma_3, I_j > I_i \quad for j \in M \cup \Gamma_2 \cup \Gamma_3 and i \in N_{18}(j) \tag{2.3}$$

where $I_i$ denotes the intensity of voxel $i$ and $N_{18}(j)$ is the 18-connected neighborhood of voxel $j$.

The first condition allows growing the mask in regions of non-increasing intensity assuming the appropriate tissue classes and the probability mask encouraging us to

grow. While in the second condition we can add a CSF voxel to the mask only if its neighbor is GM/WM and we are strictly descending an image intensity, thus the GM boundary is dilated, but only by one voxel.

The mask is allowed to dilate to voxels meeting the above provisions until the number of voxels it has added in a single sweep through the mask is less 0than 0.333% of the total volume of the MRI data set· There are voxels that still may not be included in the mask during the dilation because they are local intensity peaks, even though they are areas with an appropriate tissue class and a high probability they would have us ascending an intensity hill and so are excluded at this stage, Fig. 2(f).

To correct for these situations, a one voxel dilation, a hole filling, and a one voxel erosion are performed to our current mask, This is referred to as a modified morphological closing. The hole-filling finds the largest 6-connected component that is the background and sets all other background components to be part of the mask. We perform the dilation and erosion, simply to avoid situations where the intensity peaks may be 6-connected to the background.

The mask obtained by this algorithm is shown in 2.2(g) while the original mask is shown in 2.2(h).

Overall, The algorithm [7] that we used first performs a fast non-linear registration based on the Adaptive Bases Algorithm (ABA) [26] from multiple atlases to the subject. It also generates a tissue classification on the subject using FANTASM [25]. The registration and segmentation are combined to initialize a GM/WM region within the subject, Two morphological operations follow to ensure we have not included any of the image intensity, segmentation, and registration results as part of a dilation operation to descend the GM/WM hill to the CSF boundary.

## 2.2 Grab Cut Segmentation

The techniques designed for solving that general matting problem are effective when there is separation of foreground and background color distributions. The Grab Cut approach [6] is initially based on the handling of segmentation. They have two enhancements to the graph cuts mechanism: "iterative estimation" and "incomplete labeling" which together allow a considerably reduced degree of user interaction for a given quality of the result. This allows GrabCut to put a light load on the user, whose interaction consists simply of dragging a rectangle around the desired object. In doing so, the user is indicating a region of background, and there is no need to mark a foreground region. Secondly, it has a mechanism for alpha computation, used for border matting, whereby alpha values are regularised to reduce visible artifacts.

The Grabcut algorithm by Rother et al. [6] builds upon the idea presented by Shi and Malik. It uses a similar graph structure, where each node represents a pixel. Each node is connected to the nodes representing its pixel neighbors, and also to the foreground and background node. After performing mincut [5] on the graph, the nodes connected to the foreground node will indicate the foreground pixels, and the nodes connected to the background node will indicate the background pixels. Mincut algorithm cuts the edges with the minimum weights.

The important part of the algorithm lies in the way the weights of the edges are assigned, as that is the information that the mincut algorithm uses to partition the graph. For the rest of the discussion, we will consider *unary* weights to be the weights of the edges from each node to the foreground/background node, and *pairwise* weights to be the weights on the edges connecting neighboring pixels.

The model described by Rother et al.[6] uses Gaussian Mixture Models (GMMs) to represent the color space in the images. We have two GMMs in play, one that represents the foreground and one that represents the background. The number of components in each GMM is set such that the variability in the foreground/background can be captured appropriately. For example, if the background is composed of two colors, yellow and red, one component of the GMM would fit on the yellow pixels, while the other one will fit on the red. To initialize these GMMs, the algorithm uses K-means clustering, as it is relatively fast and clusters similar colors into a single component. This algorithm also uses a fixed number of Gaussian components, namely five, as this seems to work well on a vast number of images.

The algorithm first uses the bounding box supplied by the user to set initial foreground and background pixels. Inside the bounding box is considered to be foreground and outside is considered to be the background. Two GMMs are created by running K-Means clustering on the initial segmentation.

The algorithm then goes through an iterative process of improving the segmentation. We first assign each pixel a component from the background and the foreground GMM. This is done by simply checking which gaussian component in each of the GMMs returns the highest probability for any given pixel. Next, using our current segmentation (from the previous iteration, or from the bounding box for the first iteration) we recompute the parameters of each GMM. Basically, we take all the pixels currently segmented as foreground and update the mean and covariance of the foreground GMM, and similarly, we use the background pixels for the background GMM.

We then assign the weights to the edges of the graph as follows: The unary weights come from each of the Gaussian. They are computed as the negative of the log probability of a pixel belonging to the GMM. To reduce computational overhead, the grab

cut paper suggests that we compute the probability of only the assigned component for every pixel. Let $u_f$ represent the weight from the foreground GMM and $u_b$ represent the weight from the background GMM. The pairwise weights, $p_{(i,j)}$ are computed from the difference between the two pixels. Specifically, $p_{(i,j)} = \gamma \exp -\beta \|z_i - z_j\|^2$, where $z_i$ and $z_j$ represent the pixel indexed by $i$ and $j$. $\gamma$ here is set to 50 from empirical observations, while $\beta$ is computed from the expected difference between pairs of pixels from the image. Note that a higher similarity between the nodes would result in a bigger pairwise term. The example graph in 2.3 shows how the weights are set for each node.

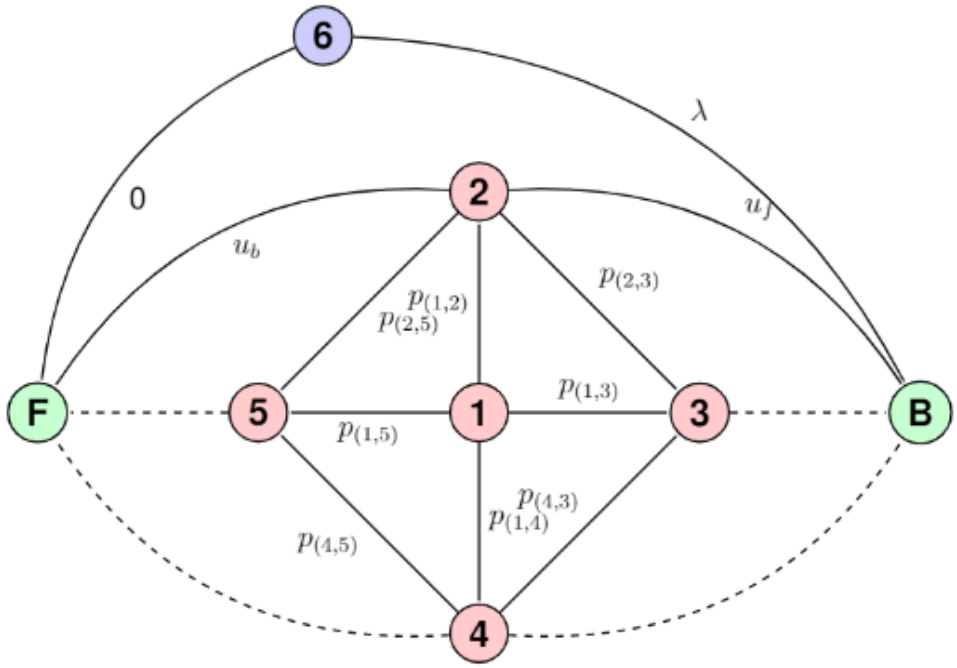



**Figure 2.3: Simplified illustration of grabcut's internal graph representation**
Red nodes indicate pixels inside the bounding box. Blue nodes represent a pixel outside the bounding box (i.e. definitely background). Green nodes represent special foreground and background nodes respectively (i.e. source and sink) [6]

For the nodes inside the bounding box i.e red nodes, a unary term from foreground GMM is assigned to the edge between a node and the background node and vice versa. This is done because the unary terms are negative log probabilities and a lower value indicated higher probability i,e if $u_b$ is lower then the node has a higher probability of being in the background and the min cut algorithm will cut $u_b$ to make the node partition with the background node. For the nodes outside the bounding box (blue nodes), we assign 0 weight to the edge to the foreground node, and $\lambda$ weight to the edge to the background node. The idea is to set $\lambda$ to a very high value so that the edge weighted 0 is always cut, and the node is always assigned as background.

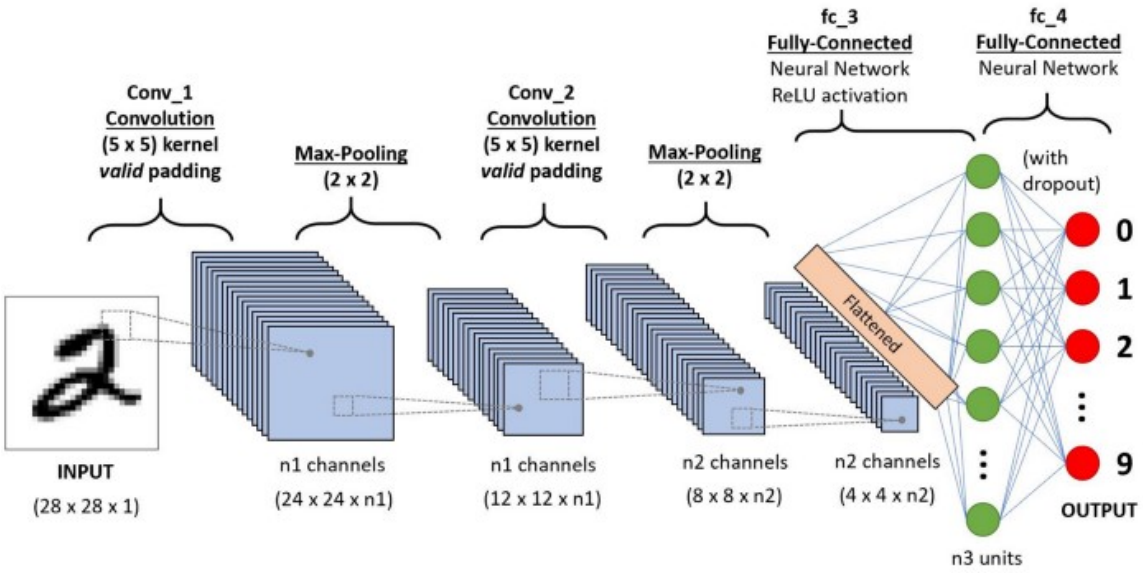
The pairwise terms propagate the information given by the unary terms. For example, if we know a node is definitely background, and a neighboring node is connected with a high pairwise weight, we would like the mincut algorithm to not cut this edge, as the neighbor is probably a part of the background.

To sum up the grab cut method, two GMMs are created by taking the user bounding box as input While the area inside the bounding box is foreground the area outside is background. A graph is created with pairwise weights and unary weights mapping each pixel node to neighbor nodes and the special foreground and background nodes. We use mincut algorithm on unary weights to classify the pixel as foreground or background. This process is continued until convergence. We then run mincut on this graph, and use the resulting partition to assign each pixel to either the foreground or the background, and re-iterate until convergence.

## 2.3 Convolutional Neural Networks

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to var-

ious aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet **??** is similar to that of the connectivity pattern of Neurons in the human brain.




An example of CNN structure [28]

**Figure 2.4: Convolutional neural network**

A ConvNet is able to capture the spatial and temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

The objective of the convolution operation is to extract high-level features such as edges from the input image. ConvNets need not be limited to only one convolutional layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level

features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the high-level features as well, giving us a network that has a wholesome understanding of images in the dataset, similar to how we would. There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in case of the former, or the Same Padding in the case of the latter.

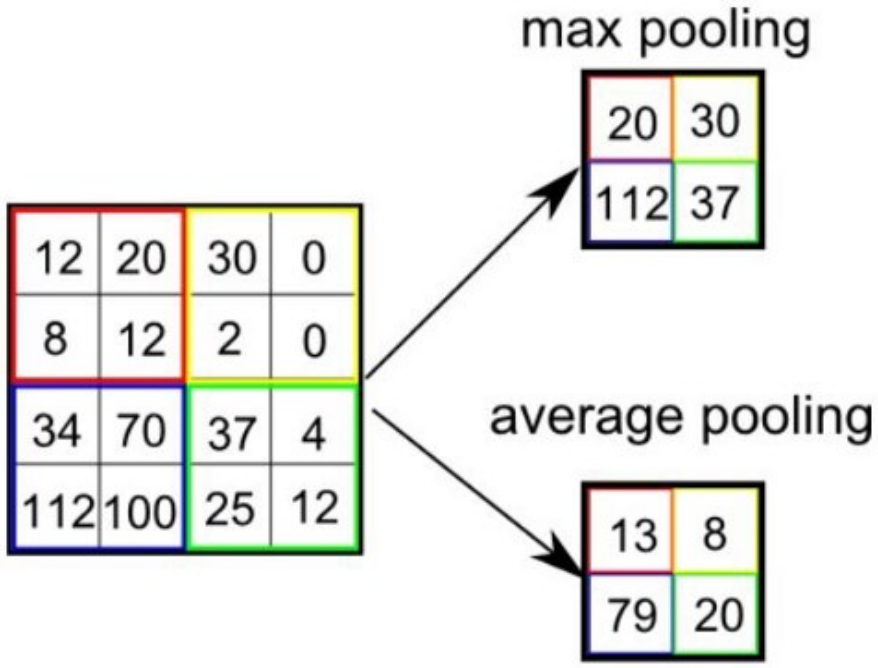



**Figure 2.5: Pooling**

Similar to the convolutional layer, the pooling layer is responsible for reducing the spatial size of the convolved feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling 2.5. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other

hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

The final output is flattened and feed it to a regular Neural Network for classification purposes. Adding a Fully-Connected layer 2.6 is a usual way of learning non-linear

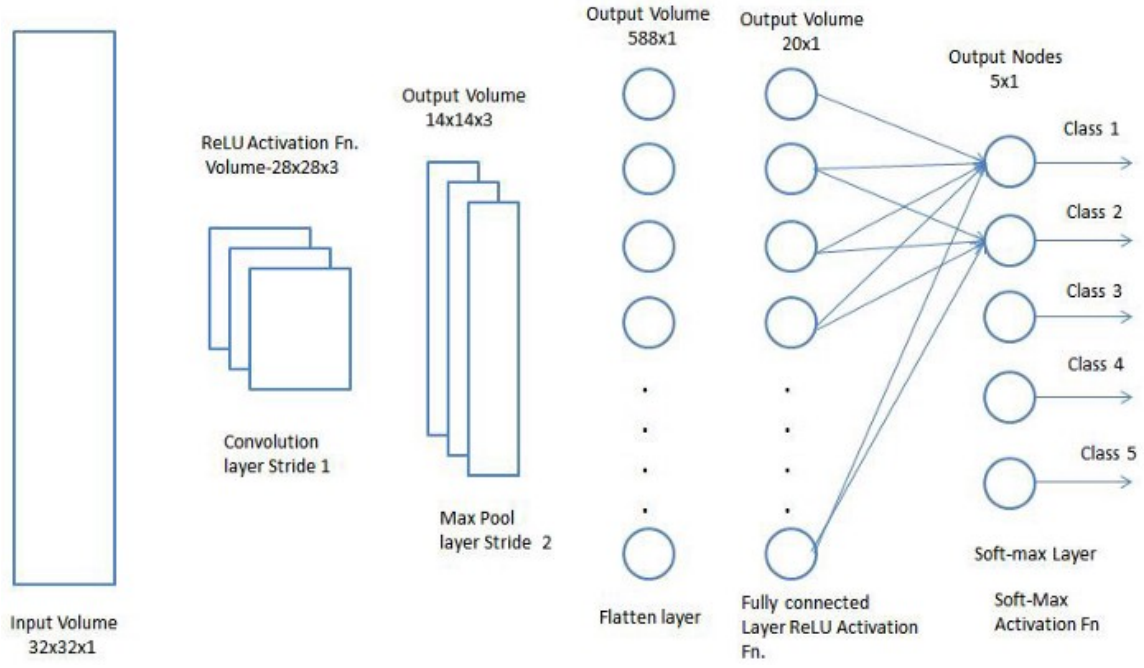



**Figure 2.6: CNN classification - fully connected layer**
CNN classification layer description [28]

combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

Now that we have converted our input image into a suitable form, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the softmax classification technique.

## 2.4 ResNet

In General, to solve complex problems we use deep neural networks with a large no of layers. The reason behind this is that these layers progressively learn complex features. In the case of recognizing features the first layers may learn to detect edges, the consequent layers may learn to identify textures, and so on. But the maximum threshold for depth with a traditional convolutional neural network (CNN) poses limitations to the number of layers that can be added. Because as the number of layers increases the accuracy decreases with traditional CNN.

This problem of training very deep networks has been alleviated with the introduction of ResNet or residual networks and these Resnets are made up from residual blocks 2.7. The core idea of ResNet is introducing a so-called "identity shortcut connection" that skips one or more layers,

it is argued by [13] that stacking layers shouldn't degrade the network performance because we could simply stack identity mappings (layer that doesn't do anything) upon the current network, and the resulting architecture would perform the same.

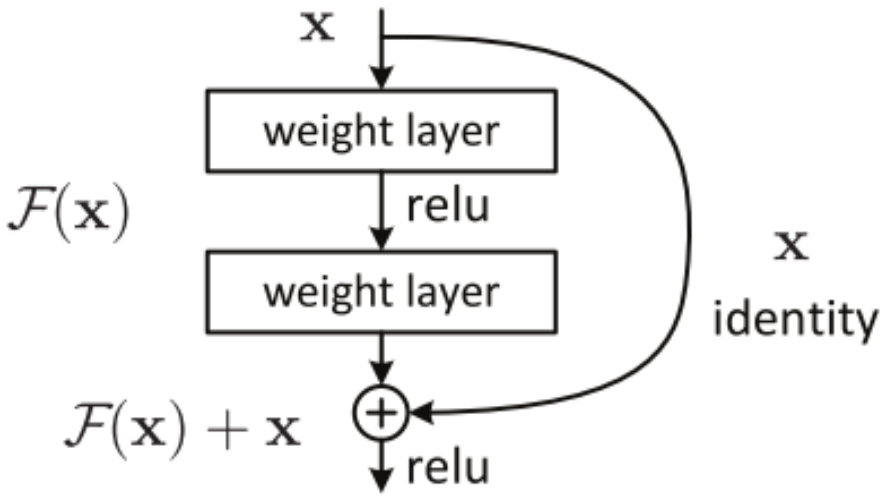



**Figure 2.7: Residual connection**

This indicates that the deeper model should not produce a training error higher than its shallower counterparts. They hypothesize that letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlying mapping. And the residual block above in 2.7 explicitly allows it to do precisely that.

ResNet was not the first to make use of shortcut connections, Highway Network introduced gated shortcut connections. These parameterized gates control how much information is allowed to flow across the shortcut. A similar idea can be found in the Long Term Short Memory (LSTM) [29] cell, in which there is a parameterized forget gate that controls how much information will flow to the next time step. Therefore, ResNet can be thought of as a special case of Highway Network. However, experiments show that Highway Network performs no better than ResNet, which is kind of strange because the solution space of Highway Network contains ResNet, therefore it should perform at least as well as ResNet. This suggests that it is more important to keep these "gradient highways" clear than to go for larger solution space.

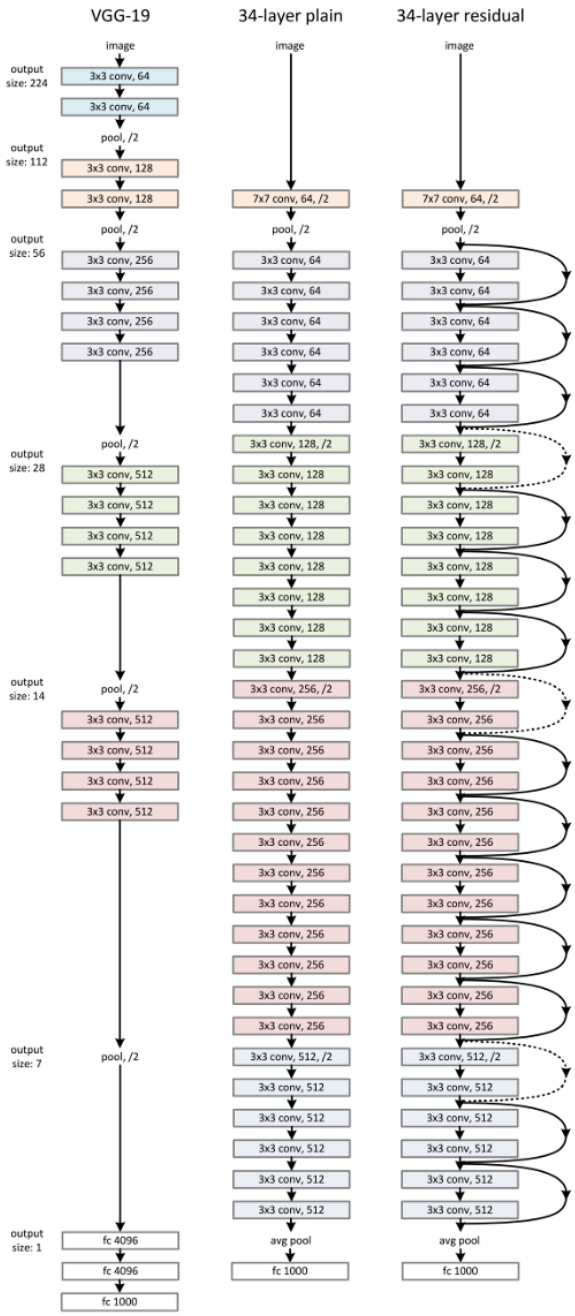



**Figure 2.8: Resnet architecture**

Following this intuition, the authors of [?] refined the residual block and proposed a pre-activation variant of the residual block [?], in which the gradients can flow through the shortcut connections to any other earlier layer unimpededly. In fact, using the original residual block in [?], training a 1202-layer ResNet resulted in worse performance than its 110-layer counterpart. Some of the variants of residual blocks are shown in 2.9.

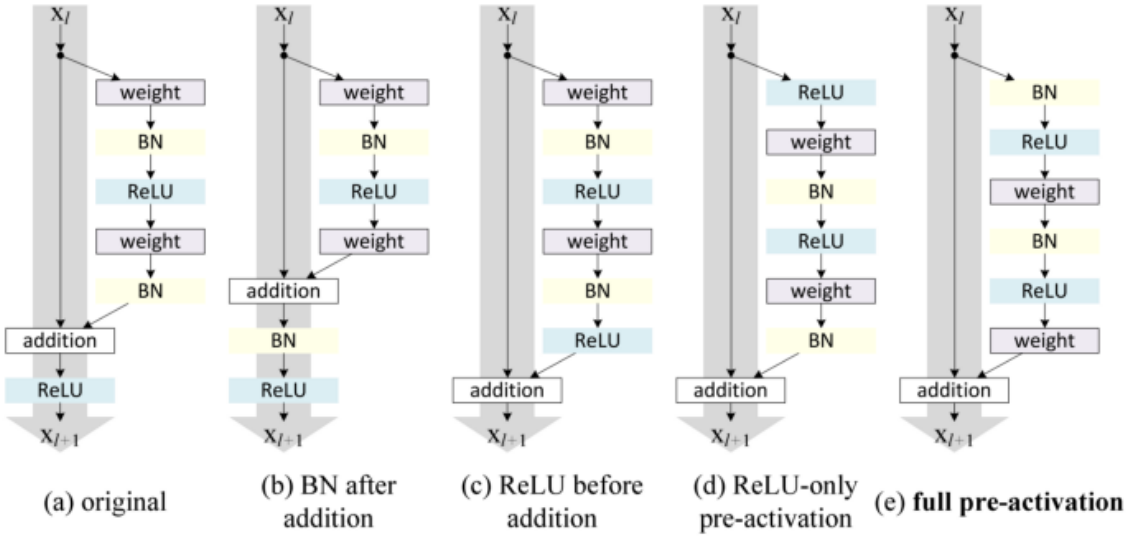



**Figure 2.9: ResNet block variants**

## 2.5 Discriminative Learning Rates

The discriminative learning rate is when you set different learning rates for different layers, usually lower learning rates at early layers and higher learning rates at later layers.

So, for the convolutional neural network, we know that the layers at the beginning are good at learning general features whereas the layers towards the end of the architecture learn specific features. When we unfreeze all the layers and are ready to update the weights, we can set the learning rate to be different for different layers.

Specifically, we want the first layer to have the lowest learning rate and the final layer to have the highest learning rate. The layers in between should have a learning rate ranging between the lowest and the highest learning rate.

The reason for the different learning rates for different layers is because layers at the beginning require very little fine-tuning since they mainly capture general features. The activation for different layers from initial to end is captured and is shown in2.10. It is observed that end layers are more activated than initial layers.

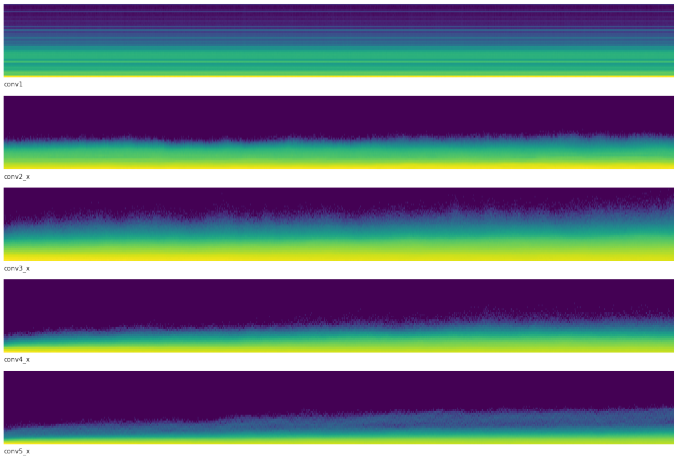


**Figure 2.10: Activations of different layers with time on x-axis**
Each plot corresponds to a single layer and the initial layer is at the top and final at the bottom [

As different layers capture different types of information they should be fine-tuned to different extents. We used a novel fine-tuning method called discriminative fine-tuning [15]. Instead of using the same learning rate for all layers of the model, discriminative fine-tuning allows us to tune each layer with different learning rates. For context, the regular stochastic gradient descent (SGD) update of a model's parameters at time step t looks like equation (2.4):

$$\theta_t = \theta_{t-1} - \eta.\Delta_\theta J(\theta) \tag{2.4}$$

where $\eta$ is the learning rate and $\Delta\theta_J(\theta)$ is the gradient with regard to the model's objective function. For discriminative fine-tuning, we split the parameters $\theta$ into $\theta_1$, . . . , $\theta_L$ where $\theta_l$ contains the parameters of the model at the l-th layer and L is the number of layers of the model. Similarly, we obtain $\eta_1$ , . . . ,$\eta_L$ where $\eta_l$ is the learning rate of the l-th layer. The SGD update with discriminative finetuning is then the (2.5) :

$$\theta_t^l = \theta_{t-1}^l - \eta^l.\Delta_{\theta^t} J(\theta) \tag{2.5}$$

It is empirically found to work well to first choose the learning rate $\eta_l$ of the last layer by fine-tuning only the last layer and using $\eta_{l-1} = \eta_l/2.6$ as the learning rate for lower layers. For adapting its parameters to task-specific features and for the model to quickly converge to a suitable region of the parameter space at the beginning of training and then refine its parameters.

Using the same learning rate (LR) throughout training is not the best way to achieve this behavior. Instead, slanted triangular learning rates (STLR) [15] is used, which first linearly increases the learning rate and then linearly decays it according to the following update schedule, which can be seen in 2.11

$$cut = [T.cut\_frac] \tag{2.6}$$

$$p = \begin{cases} t/cut, if t < cut \\ 1 - \frac{t-cut}{cut.(\frac{1}{cut\_frac-1})}, otherwise \end{cases} \tag{2.7}$$

$$\eta_t = \eta_{max}.\frac{1 + p.(ratio - 1)}{ratio} \tag{2.8}$$

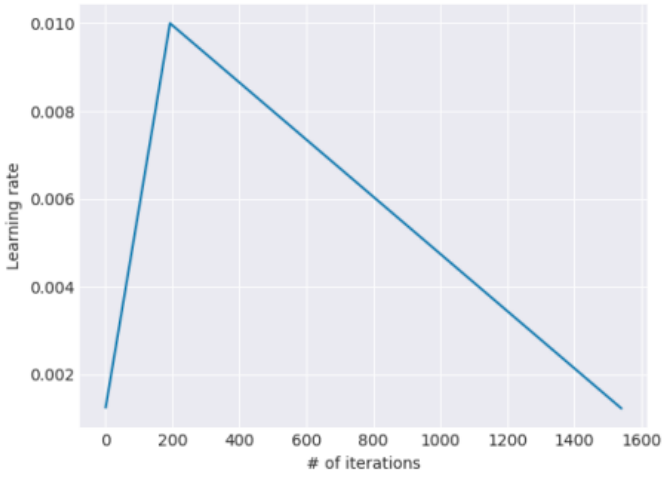



**Figure 2.11: Slanted traingular learning rate**
**The slanted triangular learning rate schedule used as a function of the number of training iterations [15]**

where T is the number of training iterations, $cut\_frac$ is the fraction of iterations we increase the LR, $cut$ is the iteration when we switch from increasing to decreasing the LR, $p$ is the fraction of the number of iterations we have increased or will decrease the LR respectively, $ratio$ specifies how much smaller the lowest LR is from the maximum LR $\eta_{max}$, and $eta_t$ is the learning rate at iteration $t$. Generally $cut\_frac = 0.1$, $ratio = 32$ and $\eta_{max} = 0.01$ is used.

The techniques used in this research are described in this chapter starting with skull stripping followed by grab cut and discriminative learning rates. In the next chapter we discuss the related works to this research and draw out some similarities and differences to our work.

Chapter 3

# RELATED WORK

This chapter describes the approaches that are related to our work. The work by Naser et al. [24] uses a combination of CNN based on U-Net to segment the tumors into different grades of glioma. The research by begum et al. [3] proposed a system with RNN and optical wavelet statistical features to perform segmentation to get the region of interest and then classification. This work also included noise removal techniques in pre-processing. Hashem et al. [12] proposed a method that uses two CNNs and a join distribution is created using NADE and the output segmentation is a combination of two CNNs. The research work [20] uses a two-level classification one into benign and malignant tumors and then malignant is classified into glioma and meningioma. They used AlexNet and GoogleNet respectively for classification. The work [34] used transfer learning on VGG by using a block-based hyperparameter tuning in which they divided the neural net into six blocks and fine-tuning is done in each block with previous blocks frozen.

## 3.1 Brain tumor segmentation and grading of lower-grade glioma using transfer learning

Glioma tumors have different grades which help in estimating the survival rates of the patients. In this work [24] they used a combination of CNN based on the U-Net to segment the tumors. They used a transfer learning-based approach with Vgg16 as the base model. This work used T1 flair images of 110 patients of Lower Grade Glioma(LGG). The model classified LGG into grade II and grade III with an accuracy

of 89% for MRI images and 95% for patients. The results achieved in this work proved that transfer learning can be used for tumor segmentation and classification. In our work, we used transfer learning for classifying different types of tumor while in this work they used it to classify into different grades of Glioma tumor. This work also used segmentation combined with classification since the Grades of the tumor mostly dependent on the shape of the tumor. In our work, we did not use segmentation to get the masks and use masks to classify because the three different types of tumors can also be differentiated based on their position, Using the masks restrict the model only to make use of the shape and texture.

## 3.2 Combining optimal wavelet statistical texture and recurrent neural network for tumor detection

This Research [3] proposed a system with a combination of optimal wavelet statistical features and recurrent neural network (RNN). The system includes noise removal, feature extraction, feature selection, classification, and segmentation. For noise removal, they used a Gaussian filter for better classification and segmentation. In Feature extraction, the number of features is reduced based on Oppositional Gravitational Search Algorithm (OGSA). The classifier uses selected features that are input into Recurrent Neural Network(RNN) to classify an image into having or not having a tumor. After classification Images having tumors are segmented to get the Region of interest with a modified growing algorithm (MRG). They achieved an accuracy of 95%.In our approach since we are using skull stripping which contains Morphological operations in the output mask the noise is not retained which is also verified by the grab cut method. The combination of Morphological Dilation and Erosion is proven in [16] to remove noise to a great extent.

### 3.3 Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE

In this paper [12] they used MRI images to train a paradigm that consists of a neural autoregressive distribution estimation (NADE) and a convolutional neural network (CNN). They have four partitions in architecture two CNNs, a distribution estimation, There are four partitions in models' architecture including two CNNs, a distribution estimator, and a fully connected network. A joint distribution is found to help feature extraction. NADE helps in identifying the brain cell placement which is useful to identify the shapes of the tumor. Similar to this method where two CNNs learn features independently and output to autoregressive model the Grab Cut Method also uses two GMMs one for tumor and the other for background. The main difference is that here the outputs of two CNNs are complemented while in Grab Cut the two GMMs one estimates the tumor pixels and the other estimates the background pixels. They used cross-entropy loss function and trained on the 3064 T1-weighted images which are in the same line with our research work. The accuracy achieved is 95 % with 6 cross-fold validation.

### 3.4 A framework for Brain Tumor segmentation and classification using Deep Learning

This work [20] proposes a framework for brain tumor detection that includes Skull stripping and tumor segmentation. They used transfer learning with two different models AlexNet and GoogleNet and implemented a hierarchy for classification. AlexNet to classify into Malignant and Benign Tumors and Google Net to classify into Glioma and Meningioma for tumors that are classified as Malignant. The preprocessing includes Data Augmentation of Rotating and Flipping. They achieved a

precision of 0.9375, recall of 1, and f-measure of 0.96 for classification in to malignant and benign. They achieved a precision of 0.95, recall of 1, f-measure of 0.9743, and accuracy of 0.9750 for classification of Malignant tumors into Glioma and Meningioma using GoogleNet. While in our method we employed only a single level of classification into Meningioma, Glioma, and Pituitary Tumors this work is similar to our work in terms of the sequence of steps. We added a step for verification of Skull Stripping using the Grab cut Method to make sure that the T1-Flair properties are retained in the output masks of skull stripping.

### 3.5 Brain tumor classification for MR images using transfer learning and fine-tuning

In this work, [34] the authors used transfer learning with pre-trained CNN as the base model. Since the dataset available in the medical analysis is generally small transfer learning will be of great advantage. They proposed a block-wise fine-tuning strategy based on transfer learning. The VGG is divided into six blocks and the fine-tuning is done from the last block and all other blocks are frozen. If in case of fine-tuning the last two blocks all other layers before them are frozen. They evaluated the proposed method on 3064 T1 weighted magnetic resonance images benchmark data set. They achieved an accuracy of 94.42% for five-fold cross-validation. Our approach also includes transfer learning but instead, we used discriminative learning for fine-tuning. Although our core idea is to retain the weights of the initial layers as they are good in recognizing the basic features we used slanted triangular learning rates distribution instead of linear learning rate distribution.

This chapter summarizes the previous approaches related to our thesis. The work by Naser et al. [24] followed segmentation to get the masks and then used masks for

classification since they are dealing with the same type of tumor but different grades. We did not follow this approach our classification is for different types of tumors. The work by Begum et al. [3] followed some noise removal techniques to clean MRI images. Since we are performing skull stripping which involves morphological operations we did not perform any additional noise removal techniques. Hashem et al. [12] proposed a method that uses two CNNs and a join distribution for segmentation which is similar to the grab cut algorithm but instead it uses GMMs to get the foreground and background pixels. The work [34] used transfer learning on Vgg19 by using a block-based hyperparameter tuning to get better results. Instead, in our work, we used discriminative learning rates for hyperparameter tuning.

Chapter 4

# METHODOLOGY

This chapter describes the methodology and system diagram that we used for the classification of brain tumors.

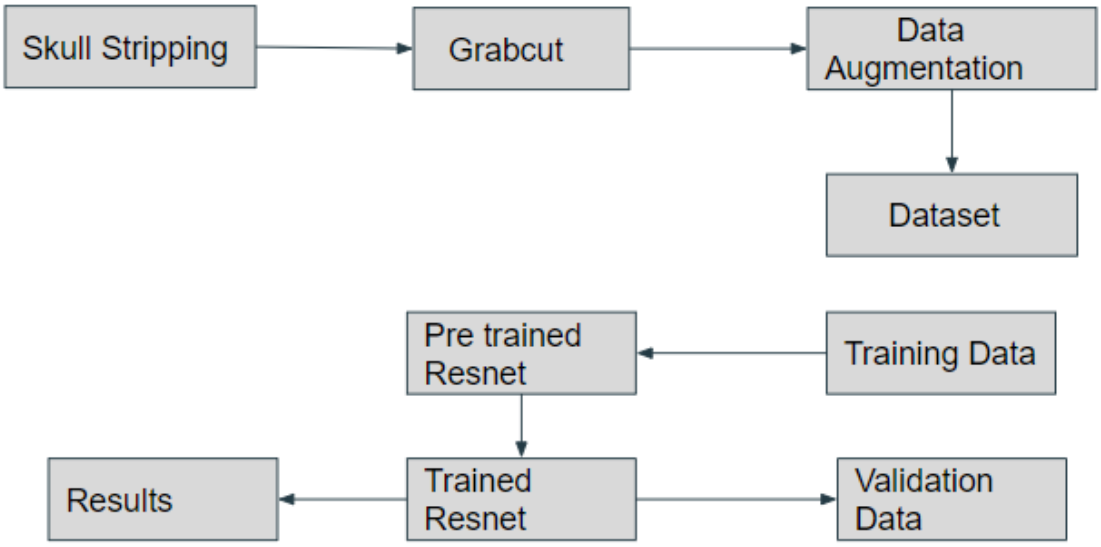



**Figure 4.1: System diagram**

## 4.1 Pre-Processing

The pre-processing involves Skull stripping to remove the skull from the MRI images, the Grab cut method to verify the mask obtained in the skull stripping, and Data Augmentation.

### 4.1.1 Skull Stripping

The skull is present around the outer area of the MRI image. To achieve better classification the skull is removed since the shape of the skull does not affect the outcome of the classification. Grey Matter (GM) and White Matter (WM) are the things we are interested in. They are connected in such a way that the point from high intensity to either endpoint is connected. The height information is represented by intensity and voxels are said to be connected if they are on the same peak and disconnected if they are on different peaks or separated by a valley. The steps in the order performed in skull stripping are

- A registration based on the Adaptive Bases Algorithm (ABA) [26] between the subject brain and atlases is performed. Atlases are a combination of an intensity image and its segmented image that the algorithm uses for registration
- The mask for each atlas is then transformed into the subject space based on the output of the registration. The mask is a representation of the cerebral cortex, cerebellum, and brain stem. We then generate a probability mask given in the subject space at voxel $i$ as, $Pi$ = (No of Masks that include i)/N, see Fig. 2.2(a).
- We also generate a tissue segmentation of the whole MRI using FANTASM [25] which has two classes $\Gamma_2$, $\Gamma_3$ an approximation of WM and GM along with dura, skull, and adipose tissue.
- The other tissue classes break down into a class made up of CSF, bone, and background ($\Gamma_1$) and another ($\Gamma_4$) dominated by the skin and adipose tissue and a small number of WM voxels.

- This initial segmentation provides a representation of the WM, and subcortical GM with a high degree of confidence.
- The segmentation and registration are combined to obtain the final mask using criteria in equation 2.1.
- This mask includes tissue from the neck and the base of the brain stem and also contains some errors in registration that led to dura and skull being included.
- To overcome this morphological erosion is performed which means the value of output pixel is the minimum of all pixels. A pixel is set to 0 if any of the neighboring pixels have the value 0. Overall this operation only keeps substantive objects. This erosion is done twice and the largest 6-connected component is found.
- Using both the outputs of registration and segmentation hill descent is performed when the voxels meet the criteria in equation 2.2 or equation 2.3 they are included.
- There are voxels that still may not be included in the mask during the dilation because they are local intensity peaks
- To correct for these situations, a one voxel dilation, a hole filling, and a one voxel erosion is performed to our current mask, The hole-filling finds the largest 6-connected component that is the background and sets all other background components to be part of the mask. We perform the dilation and erosion, simply to avoid situations where the intensity peaks may be 6-connected to the background.

The obtained mask does not contain the skull. But because of morphological operations we performed need to verify whether the masks retain the features to classify

the tumor, To verify this we use the grab cut algorithm that takes the mask and the part of the tumor mask and it should obtain the complete tumor mask to consider the mask in the dataset.

### 4.1.2 Grab Cut

Grab cut is used in our system to verify whether the mask obtained from skull stripping contains features for the tumor to be detected. The Grabcut algorithm by Rother et al. [6] builds upon the idea presented by Shi and Malik. It uses a similar graph structure, where each node represents a pixel. Each node is connected to the nodes representing its pixel neighbors, and also to the foreground and background node. The idea is that after performing min-cut on the graph, the nodes connected to the foreground node will indicate the pixels in the foreground, and the nodes connected to the background node will indicate the pixels in the background.

It uses Gaussian Mixture Models to represent the color space in the images. We have two GMMs in play, one that represents the foreground and one that represents the background. The number of components in each GMM is set such that the variability in the foreground/background can be captured appropriately. This algorithm uses K-means, as it is relatively fast and clusters similar colors into a single component. It also uses only five gaussian components.

The Grab cut based tumor detection Algorithm is described as:

- Step1-MRI Image is taken and segmented into tumor as foreground and all other parts as the background image.
- Step 2-Assign label = 1 for tumor region obtained from bounding box and label = 0 for non-tumor region.

- Step 3-Two sets of GMM parameters mean, co-variance are estimated using Gaussian components model one for tumor and the other for background region.
- Step 4-Apply Expectation-Maximization algorithm to get better iterative segmentation by energy minimization function.
- Step 5-Tumor region is extracted from background image by taking the color of foreground pixel and assigning those pixels around the boundary extraction to reduce the blurriness in pixel.

We verify that the features to detect the tumor are still retained if the output of the grab cut matches the original tumor mask. In this case, the image is retained, or else the image is considered to be high in noise or lost its features to detect the tumor and removed from the dataset.

### 4.1.3 Dataset and Augmentation

The dataset used is the 3064 -T dataset [8] which contains T1 flair MRI images. The tumor is of 3 different classes namely Meningioma, Glioma, and Pituitary. The dataset in Mat format is converted to the .npy array list and the data is transformed into 8 images of sets (0-45 degrees),(45-90 degrees) ... (325-360 degrees). The labels are modified accordingly for the training and validation and test set. After augmentation the no of MRI images in dataset is increased by 8 times.

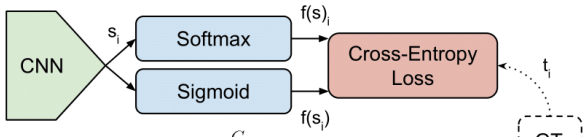
## 4.2 Transfer Learning and Training

We used ResNet50 [13] as the base model. ResNet is trained with Imagenet [9] dataset. The model is initially trained by freezing all the layers except the last 2 layers with

the training data. Hyperparameter tuning is done to find the optimal learning rate to train with. After training the last two layers then the model is unfrozen and hyperparameter tuning is done once again and the layers are trained with discriminative learning rates to make sure that the initial layers do not change by much while the last years change the most. The Slanted triangular learning rates described in chapter 2 are used for hyperparameter tuning. This is done based on the assumption that the pre-trained ResNet is better at recognizing initial features and large data is not required for training.

### 4.2.1 Loss Function

The loss function we used is Cross-entropy. Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events. Entropy is the number of bits required to transmit a randomly selected event from a probability distribution. A skewed distribution has low entropy, whereas a




$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

$$CE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1 - t_1) log(1 - f(s_1))$$

**Figure 4.2: Cross entropy**

distribution where events have equal probability has a larger entropy. Cross-entropy builds upon the idea of entropy from information theory and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution. In this case, one distribution is the final dense layer output of the input image and the other is the final dense layer output of the target image.

Chapter 5

# EXPERIMENTAL SETUP

## 5.1 Dataset

This brain tumor dataset obtained from Figure Share [8] contains 3064 T1-weighted contrast-enhanced images from 233 patients with three kinds of brain tumor: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). Due to the file size limit of the repository, the whole dataset is split into 4 subsets, and archive in 4 .zip files with each .zip file containing 766 slices. The 5-fold cross-validation indices are also provided.

This data is organized in matlab data format. Each file stores a struct containing the following fields for an image:

- cjdata.label: 1 for meningioma, 2 for glioma, 3 for pituitary tumor
- cjdata.PID: patient ID
- cjdata.image: image data
- cjdata.tumorBorder: a vector storing the coordinates of discrete points on tumor border. For example, [x1, y1, x2, y2,...] in which x1, y1 are planar coordinates on tumor border. It was generated by manually delineating the tumor border. So we can use it to generate binary image of tumor mask.
- cjdata.tumorMask: a binary image with 1s indicating tumor region

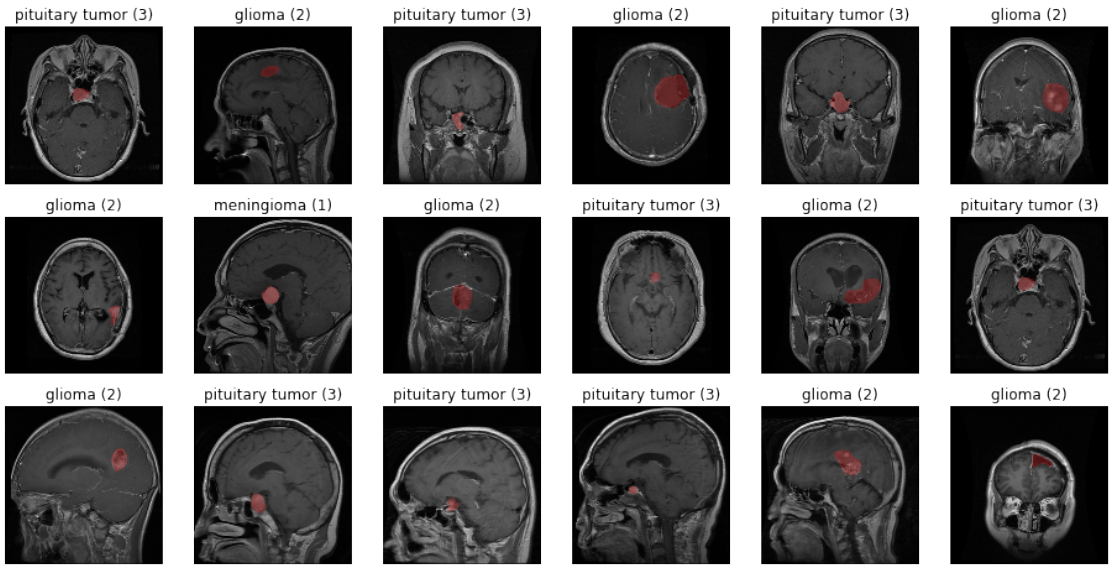



**Figure 5.1: Dataset in different axial planes**

## 5.2 Baselines

The Dataset is divided into Training, Validation, and Test Set of 70%, 15%, and 15%. The Accuracy and F1 score are calculated. The accuracy and loss graphs are plotted for training and validation data, A classification report is generated with macro average and weighted average of precision, recall, and f1 score. The accuracy and f1 score are compared with other research works of Afshar et al. [**?**], Vimal Kurup et al. [21], Srinivasan et al.[1], Swati et al. [35] and Hashemzehi et al. [12] on the same dataset.

## 5.3 Hardware Specifications

The hardware specifications are

- 16 GB DDR4 RAM;

- 12 GB NVidia Tesla K40 GPU
- 50 GB of Non-Persistent Storage

# Chapter 6

# RESULTS

The results obtained were presented in this chapter. The results are discussed and interpretations from the results are made. The results are compared with the other works in 6.1 and discussion about results are done in 6.2.

The Dataset is divided into Training, Validation, and Test Set. The training data is 70%, Test data is 15% and validation data is 15%. The output classes are Meningioma, Glioma, and Pituitary tumors. The precision (equation 6.1), recall (equation 6.2) and f1 score (equation 6.3) are calculated for each target class. The overall accuracy is calculated for the training test and a validation dataset is also obtained.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{6.1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{6.2}$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.3}$$

After performing data augmentation the samples in each train, validation, and test data are 17152, 3680, and 3680 respectively. The loss function used is Cross-Entropy Loss and the optimizer used is Stochastic Gradient Descent(SGD) with a momentum of 0.9. Hyperparameter tuning for learning rate is performed to find that 3e-4 is a

good learning rate. This is used as the last layer's learning rate and the remaining layers learning rate is obtained by dividing it by 2.7. The batch size used is 4.

The precision, recall, and f1 score for each of the classes is reported in the 6.1. The F1 score for the 3 target classes is reported as 0.93, 0.97, 0.97 respectively for the test dataset. The accuracy for Train, Validation, and Test set is represented in 6.2 which is 98.83 percent for training data 95.26 percent for validation data, and 95.18 for Testing data.

There could be a possibility that the test set contains mostly a single class and our model might be biased towards that and predicts the same class for most of the inputs. To prove that the test set is unbiased and our model is not predicting the same class for all the inputs. We calculated macro average and the weighted average for precision, recall, and f1-score in 6.1 both have the same f1 score that is 0.96 which implies that the dataset is unbiased and all the classes are fairly weighted.

Clasification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.93 | 0.93 | 902 |
| 1 | 0.97 | 0.97 | 0.97 | 1713 |
| 2 | 0.97 | 0.97 | 0.97 | 1065 |
| accuracy | | | 0.96 | 3680 |
| macro avg | 0.95 | 0.96 | 0.96 | 3680 |
| weighted avg | 0.96 | 0.96 | 0.96 | 3680 |

**Figure 6.1: Classification Report**

We also plotted the loss graph for validation and training dataset in 6.4. The loss plot indicates that the training loss decreased as the number of epochs increased while the validation loss went up and down. From the validation loss graph, we can see

```
Epoch 10 Batch 2144
Accuracy: 98.83 %  Loss: 0.0012  Duration: 8.69 minutes
Validation Accuracy 95.26 Validation Loss: 0.0004
```

**Figure 6.2: Accuracy**

that we did not settle for local minima for validation loss, trained a little further. Overfitting occurs if validation loss is increasing while the training loss is decreasing. We imply that overfitting does not occur from the plot as both the training and validation loss are decreasing before converging. We also plotted the accuracy plot in 6.3 which conveys the same that both validation and training accuracy is increasing at the same time. If validation accuracy is decreasing at the cost of increasing in training accuracy in that case we can say overfitting occurs.

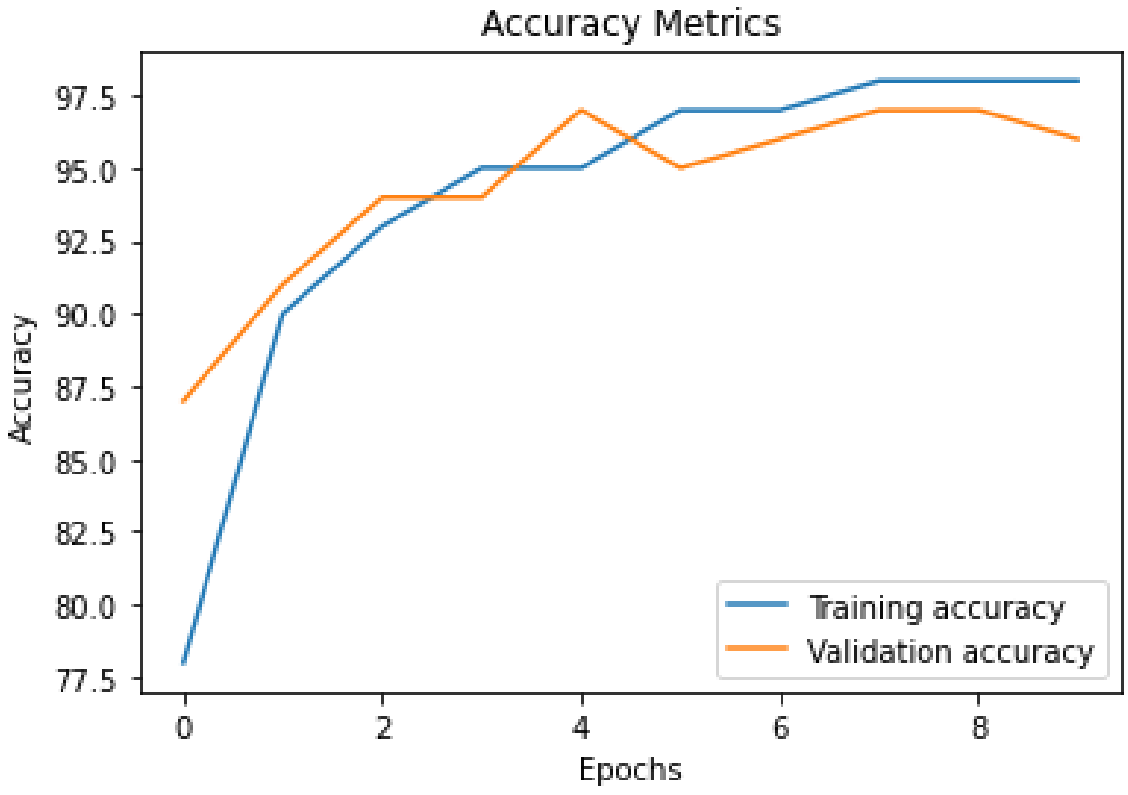



**Figure 6.3: Accuracy plot**

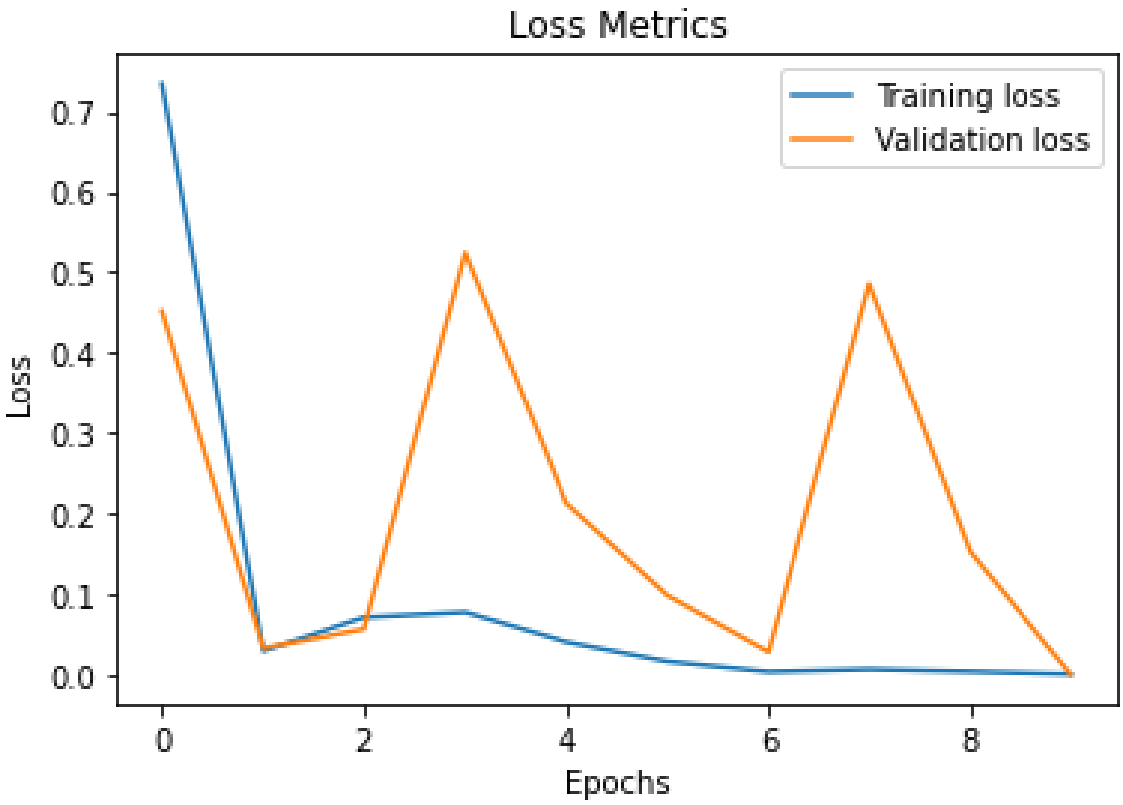



**Figure 6.4: Loss plot**

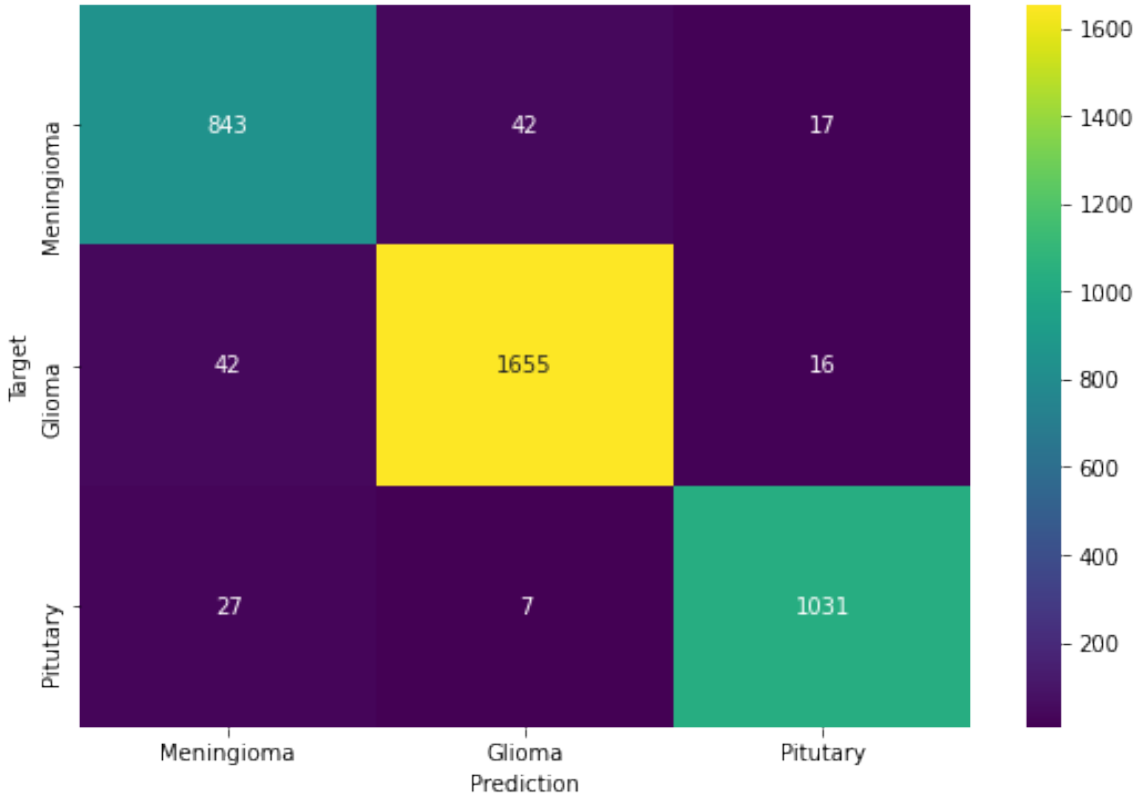



**Figure 6.5: Confusion matrix**

The Confusion matrix 6.5 is also plotted to check for the misclassified labels and correctly classified labels. class 1 is correctly labeled for 843 instances, class 2 is correctly labeled for 1655 instances and 1031 instances of class 3 are classified correctly. It can be observed in the scale that the misclassified instances are comparatively low in the case of Pituitary tumors because their spatial distribution is very constrained and limited in comparison to Meningioma and Glioma which have very wide spatial distribution.

## 6.1 Comparision

The results obtained using this method are compared with recently published papers in 6.1. In comparison with work by Hashemi et al.[12] that used NADE and CNN,

| Results Comparision | | | |
|---|---|---|---|
| Reference | Data Division | Accuracy | F1 score |
| Afshar [2] | Not stated | 86.56 | NA |
| Vimal Kurup [21] | 80 % in training and 20 % in test | 92.60 | 0.91 |
| Srinivasan [1] | 75% in training and 25 % in test | 93.30 | NA |
| Hashemezehi et al. [12] | Not stated | 95% | 0.94 |
| Swati et al. [35] | Not stated | 94.42% | 0.91 |
| Proposed Method | 70 % in training ,15% in validation and 15% in test | 95.26% | 0.96 |
| | | | |

**Table 6.1: Comparision of results**

our transfer learning method performed better because of the size of the dataset. We performed data augmentation which increased the size of the dataset and exposed our model to various possible shapes of the tumor. Transfer learning models are good at detecting low-level features and require less data compared to non-transfer learning-based approaches. In comparison with work by Swati et al. [35], even though both of the works used transfer learning approach we performed skull stripping which eliminated features of the skull that are not useful to detect tumors, and also used discriminative learning instead of block-based hyperparameter tuning which helped us in getting better results and also achieved the results with fewer epochs. The work by Afshar et al. [2] uses a capsule network to first segment the region of interest and then detect the tumor. Even though they achieved better accuracy compared to other CNN-based models the transfer learning approaches in general performed better.

## 6.2 Discussion

The main reason for misclassification is because of variability in brain tumor sizes and locations of tumor tissues belonging to Meningioma and Glioma brain tumor classes.

The similarity of the distribution in sizes across the data set of Meningioma and Glioma tumor classes is also an added reason. This collectively indicates the difficulty any deep learning neural network would have distinguishing between Meningioma and Glioma tumor classes based on size and location alone in this data set. In addition, transfer learning approaches perform better in the case of small datasets. It is because of the base model's ability to detect the low-level features well and we only need to train for high-level features. In addition methods like skull stripping and data augmentation seems to provide better results as the model will be exposed to fewer unimportant features and different shapes of tumors. However, there are many improvements that can be made to our existing methodology which pointed in future works. The future works are discussed in the next chapter.

# Chapter 7

# FUTURE WORKS

The real-world problems require proper addressing in the domain of medical image analysis that has been external to the domain of computer vision. The computational complexity in handling MR image modalities at a single instance is the major challenge in imaging and pre-processing techniques. During MR image scan for diagnosis correct tuning of parameters and motion applied to images, are areas that can be explored in the future. The time taken for processing and segmenting is considerably high which makes it difficult to process in real-time situations. The classification using transfer learning while proven effective and takes fewer resources to train it has some bias because of the datasets that are used to pre-train the models. This should be addressed in future works to make the system more efficient. The current methods that deal with the noise in the MRI images are also responsible for reducing the features in the MRI images. Therefore models with better Noise removal techniques that don't flatten the images can further increase the model's accuracy. Techniques like semi-supervised learning can be used to augment the size of the dataset.

# Chapter 8

# CONTRIBUTION

In this work the main contributions are:

- A system that classifies tumors into 3 different types with more accuracy than current non-transfer learning methods.
- Provide sufficient evidence that transfer learning can be used for domains like Medical images even though the model is pre-trained with data that does not contain the domain.
- Usage of discriminative learning rates in case of transfer learning to make more efficient models and makes it easier to train with fewer resources.
- Classification can be more effective when appropriate pre-processing steps like grab cut and skull stripping methods are paired to achieve better results.

# Chapter 9

# CONCLUSION

The proposed methodology successfully segments the MRI images into three tumors with an accuracy of 95.18%. In comparison with the other recently published works in which the models are trained from the scratch, our model was built using transfer learning using Resnet50 as a base model and trained with discriminative learning rates performed better. We used registration and segmentation-based skull stripping to remove the skull portion and the grab cut method to verify whether the tumor features are retained after removing skull data. After training for 10 epochs we achieved 95.26 % accuracy for the validation and 95.18 % for the test set and 98.86 % for the training set and surpassed those previous models. The f1 score achieved is 0.96 macro average. For individual classes f1 score are 0.93, 0.97, 0.97 respectively. Even though the developed system classifies the tumor with reasonably good accuracy there is still a need for medical experts to confirm the results as there is a probability of error. This system can be used for preliminary analysis for brain tumor detection before it can be sent to the experts. In comparison, with the accuracy of the medical experts in diagnosing the tumor which is found to be 69% in a survey [23] the proposed system performs better. The model can be furthermore trained or additional data can be used to train to increase the accuracy as it is not close to overfitting.

# BIBLIOGRAPHY


[1] Development of deep learning algorithms for brain tumor classification using GLCM and wavelet packets. *Arxiv*.

[2] P. Afshar, K. N. Plataniotis, and A. Mohammadi. Capsule networks for brain tumor classification based on MRI images and course tumor boundaries. *arXiv preprint arXiv:1811.00597*, 2018.

[3] S. S. Begum and D. R. Lakshmi. Combining optimal wavelet statistical texture and recurrent neural network for tumour detection and classification over MRI. *Multimedia Tools and Applications*, pages 1–22, 2020.

[4] J. Bernal, K. Kushibar, D. S. Asfaw, S. Valverde, A. Oliver, R. Martí, and X. Lladó. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial Intelligence in Medicine*, 95:64 – 81, 2019.

[5] N. Bhardwaj, A. M. Lovett, and B. Sandlund. A simple algorithm for minimum cuts in near-linear time. *arXiv preprint arXiv:1908.11829*, 2019.

[6] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, pages 428–441, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[7] A. Carass, M. Wheeler, J. Cuzzocreo, P.-L. Bazin, S. Bassett, and J. Prince. A joint registration and segmentation approach to skull stripping. pages 656–659, 01 2007.

[8] J. Cheng. Brain tumor dataset. *figshare*, Apr 2017.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[10] A. Gumaei, M. M. Hassan, M. R. Hassan, A. Alelaiwi, and G. Fortino. A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. *IEEE Access*, 7:36266–36273, 2019.

[11] P. P. Gumaste and V. K. Bairagi. A hybrid method for brain tumor detection using advanced textural feature extraction. *Biomedical and Pharmacology Journal*, 13(1):145–157, 2020.

[12] R. Hashemzehi, S. J. S. Mahdavi, M. Kheirabadi, and S. R. Kamel. Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE. *Biocybernetics and Biomedical Engineering*, 40(3):1225–1232, 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2015.

[14] K.-S. Hong, M. J. Khan, and M. J. Hong. Feature extraction and classification methods for hybrid fNIRS-EEG brain-computer interfaces. *Frontiers in human neuroscience*, 12:246, 2018.

[15] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[16] N. Jamil, T. M. T. Sembok, and Z. A. Bakar. Noise removal and enhancement of binary images using morphological operations. In *2008 International Symposium on Information Technology*, volume 4, pages 1–6, 2008.

[17] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61 – 78, 2017.

[18] P. Kanmani and P. Marikkannu. MRI brain images classification: a multi-level threshold based region optimization technique. *Journal of medical systems*, 42(4):1–12, 2018.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[20] S. Kulkarni and G. Sundari. A framework for brain tumor segmentation and classification using deep learning algorithm. *International Journal of Advanced Computer Science and Applications*, 11, 01 2020.

[21] R. Kurup, V. Sowmya, and K. P. Soman. Effect of data pre-processing on brain tumor classification using capsulenet. In *ICICCT - 2019*, 2019.

[22] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim. Deep learning in medical imaging: general overview. *Korean journal of radiology*, 18(4):570, 2017.

[23] D. Lu, N. Polomac, I. Gacheva, E. Hattingen, and J. Triesch. Human-expert-level brain tumor detection using deep learning with data distillation and augmentation. *Arxiv*, 2020.

[24] M. A. Naser and M. J. Deen. Brain tumor segmentation and grading of lower-grade glioma using deep learning in mri images. *Computers in Biology and Medicine*, 121:103758, 2020.

[25] D. L. Pham. Robust fuzzy segmentation of magnetic resonance images. In *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*, pages 127–131, 2001.

[26] G. K. Rohde, A. Aldroubi, and B. M. Dawant. The adaptive bases algorithm for intensity-based nonrigid image registration. *IEEE Transactions on Medical Imaging*, 22(11):1470–1479, 2003.

[27] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23:309–314, 08 2004.

[28] S. Saha. A comprehensive guide to convolutional neural networks. *Towards Datascience*, 2018.

[29] J. Schmidhuber and S. Hochreiter. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

[30] M. Sharif, J. Amin, M. Raza, M. Yasmin, and S. C. Satapathy. An integrated design of particle swarm optimization (pso) with fusion of features for detection of brain tumor. *Pattern Recognition Letters*, 129:150 – 157, 2020.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Arxiv*, 2015.

[32] P. Siva Raja and A. V. rani. Brain tumor classification using a hybrid deep autoencoder with bayesian fuzzy clustering-based segmentation approach. *Biocybernetics and Biomedical Engineering*, 40(1):440 – 453, 2020.

[33] H. Sugimori and M. Kawakami. Automatic detection of a standard line for brain magnetic resonance imaging using deep learning. *Applied Sciences*, 9(18), 2019.

[34] Z. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmad, and J. Lu. Brain tumor classification for mr images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics*, 75:34–46, 05 2019.

[35] Z. N. K. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed, and J. Lu. Brain tumor classification for MR images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics*, 75:34 – 46, 2019.

[36] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[37] M. Talo, U. B. Baloglu, Özal Yıldırım, and U. Rajendra Acharya. Application of deep transfer learning for automated brain abnormality classification using MR images. *Cognitive Systems Research*, 54:176 – 188, 2019.

[38] G. S. Tandel, M. Biswas, O. G. Kakde, A. Tiwari, H. S. Suri, M. Turk, J. R. Laird, C. K. Asare, A. A. Ankrah, N. Khanna, et al. A review on a deep learning perspective in brain cancer classification. *Cancers*, 11(1):111, 2019.

[39] K. Wang, M. Savva, A. X. Chang, and D. Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[40] A. Çinar and M. Yildirim. Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. *Medical Hypotheses*, 139:109684, 2020.