

# Brain Tumors Detection By Using Convolutional Neural Networks and Selection of Thresholds By Histogram Selection

Kasiful Aprianto

BPS - Statistics Indonesia, Jl. Dr. Sutomo 6-8, Jakarta, 10710, Indonesia

*Email: apriantokasiful@gmail.com*

## Abstract

Brain tumors in medical images have a high diversity in terms of shape and size. Some of the data found a form between the tumor tissue and normal tissue, whereas knowing the tumor's profile and characteristics becomes a crucial part of searching. By using machine learning capabilities, where machines are given several variables and provide decisions to a certain degree, they have broadly given decisions that support subject matter in making decisions. This study applies the threshold selection method using histogram selection on CT scan data, while the appropriate threshold selection method selects the tumor position accordingly. Furthermore, the Convolutional Neural Network (CNN) is used to classify whether the selected image is a tumor or not. Using CT scan data and calculated experiments, this algorithm can finally be approved and given a brain classification with an accuracy of 75.42 percent.

**Keywords:** *Convolutional Neural Network, Histogram Selection, Machine Learning*

## 1. Introduction

The medical image becomes a remarkable breakthrough in medicine in detecting a condition in a patient's limbs. Some activities include the detection of tumors or cancer. By using medical images, the resulting image will provide an overview of the visible form so that it can determine the patient needs further treatment. Viewing medical images up to now is still a challenge for users to determine the tumor's position or classify the tumor in the brain because there is still a possibility of the shape and color of the resulting image similar to normal tissue.

With medical images, health workers can analyze and determine the position of tumors in the brain. However, the manual method requires a long time and a severe challenge to CT scans produced in routine medical activities. For this reason, experiments need to be conducted on ways that can help provide an automatic classification as a weicher in making faster decisions.

CNN for image classification in several studies was able to provide good results. Some studies include various plants on resolution images with an accuracy of 82 percent [1]. It is also used in the fruit image classification model [2], facial recognition [3], to make automatic vehicles using CNN [4]. This method is then performed a brain tumor classifica-

tion experiment using CNN. Previous research has conducted several methods to detect brain tumors. Some of them are using classification techniques, such as K-Nearest Neighbor [5], ANN [6], SVM [7], and particle swarm optimization (PSO) [8]. In this research, the preprocessing of data is done first by making a segmentation using histogram selection, then forming a model using CNN. Histogram selection works by creating the best segmentation to determine the most likely position to occur tumors [9]. CNN is given afterward to determine whether the detected is a tumor or healthy tissue.

## 2. Methodology

This research consists of several stages, namely: collecting data, converting images into grayscale, determining the position of the head, segmenting images, training data with CNN, and testing.

### 2.1. Collecting Data

The data used are data sourced from Kaggle [10] with the title MRI Brain Image for Brain Tumor Detection. The dataset consists of 96 normal brain images and 157 brain images containing tumors with different shapes and positions.

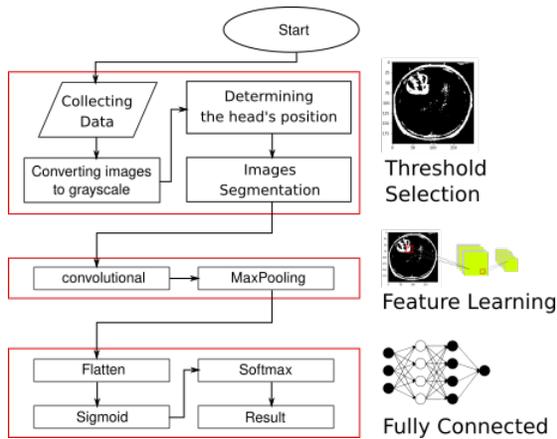


Fig. 1. Proposed research method

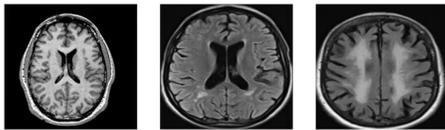


Fig. 2. The example of normal brain dataset

## 2.2. Converting to Grayscale and Determining the head's positions

Images that have been processed into grayscale, then the head position is determined for each image. Each image has a variety of sizes and positions so that the head size and position are uninformed by removing the black sides on the left, right, front, and back of the head. After that, the resolution is resized to 200 x 200 pixels. It is performed by using some computation. The head's position is determined as

## 2.3. Making the image segmentation

Based on research [9], images that have been processed as in point 2.3 are further divided into four segments, namely in the upper right, upper left, lower right, and lower left. After that, the average standard deviation for the four regions is calculated. Then calculate the average of each segment and add up the values. It aims to find the right threshold. If the value is greater than the threshold, it is marked with white, while others are marked with black. This clustering technique is used to determine the position most likely to occur brain tumors in the image. Threshold determination follows the provisions in research [9], which refer to the **Algorithm 2** on the table below.

---

### Algorithm 1 Determine head position

---

**Require:**  $rowDel = []; colDel = []; i = 0; img$

**Ensure:** collecting the top of black image

```

i = 0
while i < len(img) do
  if sum(img[i] == 0) then
    append(rowDel, i)
  else
    break
  end if
  i = i + 1
end while

```

**Ensure:** collecting the bottom of black image

```

i = 0
while i < len(img) do
  if sum(img[len(img) - 1 - i] == 0) then
    append(rowDel, len(img) - 1 - i)
  else
    break
  end if
  i = i + 1
end while

```

**Ensure:** delete all selected rows

$img = img[rowDel]$

**Ensure:** collecting the left of black image sides

```

i = 0
while i < len(img[0]) do
  if sum(img[:, i] == 0) then
    append(colDel, i)
  else
    break
  end if
  i = i + 1
end while

```

**Ensure:** collecting the right of black image sides

```

i = 0
while i < len(img[0]) do
  if sum(img[:, len(img[0]) - 1 - i] == 0) then
    append(colDel, len(img[0]) - 1 - i)
  else
    break
  end if
  i = i + 1
end while

```

**Ensure:** delete all selected rows

$img = img[:, colDel]$

---

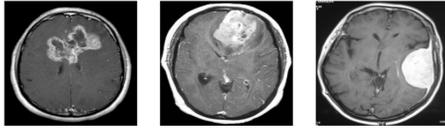


Fig. 3. The example of normal brain dataset

---

**Algorithm 2** Threshold Selection for Converting Image to Grayscale

---

**Require:**  $\alpha, \sum \bar{x}, threshold$

**if**  $\alpha < 40$  **then**

**if**  $\sum \bar{x} \geq 100$  **then**

$threshold = \sum \bar{x}$

**else if**  $\sum \bar{x} \leq 100$  **then**

$threshold = 0.5 * \sum \bar{x}$

**end if**

**else if**  $\alpha \geq 40$  **then**

**if**  $\sum \bar{x} < 65$  **then**

$threshold = 2.5 * \sum \bar{x}$

**else if**  $65 \leq \alpha < 90$  **then**

$threshold = 2 * \sum \bar{x}$

**else if**  $90 \leq \alpha < 120$  **then**

$threshold = 1.5 * \sum \bar{x}$

**else if**  $120 \leq \alpha < 175$  **then**

$threshold = \sum \bar{x}$

**else if**  $175 \leq \alpha < 250$  **then**

$threshold = \sum \frac{\bar{x}}{1.5}$

**else if**  $250 \leq \alpha < 360$  **then**

$threshold = \sum \frac{\bar{x}}{2}$

**else if**  $360 \leq \alpha < 450$  **then**

$threshold = \sum \frac{\bar{x}}{2.5}$

**else if**  $\alpha > 450$  **then**

$threshold = \sum \frac{\bar{x}}{3}$

**end if**

**end if**

---

## 2.4. Convolutional Neural Network

Convolutional Neural Network is the development of a multilayer perceptron, which is inspired by the neural network of living things [11]. CNN is part of deep learning, a branch of machine learning that can work on computers to work like humans, like computers that learn from the training process (Deng and Yu) [12]. In general, the dataset used to classify using a convolutional neural network has three layers, namely width, height, and depth. This study will use two layers, namely width, and height only because the data to be processed does not have a color depth. Broadly speaking, the CNN stage consists of Feature Learning and Fully Connected. This study uses the CNN architecture as follows: (1) convolutional as many as 50 nodes, then fully

connected with (2) 100 nodes of relay layer, (3) sigmoid of 100 nodes, and (4) softmax of 2 nodes.

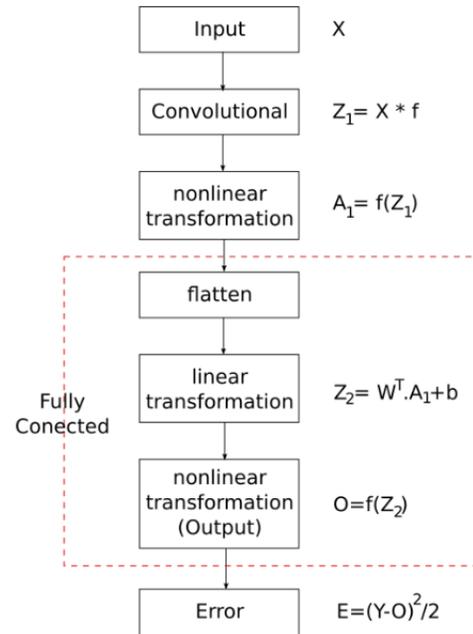


Fig. 4. The CNN's Workflow

**2.4.1. Convolutional.** In Feature learning, the computational process that occurs changes the image obtained into a collection of numbers that represent the image. The process consists of 2 stages, namely Convolutional and Pooling. At this stage, each image will be filtered. A filter is an  $(n \times n)$  matrix that is used to extract images. The next filter will be weight as training material. The filter is shifted to all parts of the image, then matrix multiplication operations are finally produced. This output will henceforth be a feature map. In comparison, the number of filter shifts can be done as much as every 1 pixel or more.

The formula for calculating convolution can be seen as shown as the image above, which can be written as:

$$Z_1 = X * f \quad (1)$$

where the  $Z_1$  value is convolutional (figure 4), the sum of each multiplication between input  $X$  and filter  $F$ . The multiplication technique, as shown above (figure 5), it shows that the filter has a fixed size. Like the picture above, the filter has  $2 \times 2$  pixels. Then, to calculate the value of  $Z$ , the filter is moved to each row and column, multiplied, adding all of them together, and it will produce output that has the size as shown below:

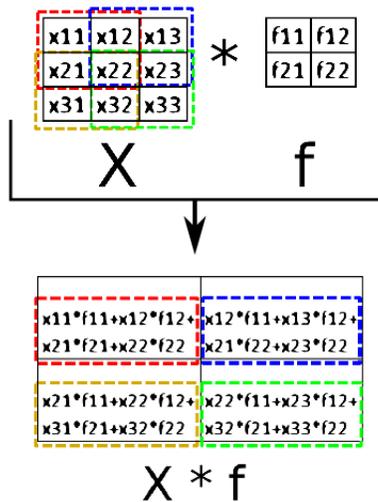


Fig. 5. Example of how convolution works on CNN

$$Dim_{output} = [(n_{row} - f_{col} + 1), (n_{col} - f_{col} + 1)] \quad (2)$$

The next stage is pooling. The pooling stage is the same as the convolutional stage. Still, the application is made after convolutional. It is not part of the weight that will be trained because it is only reducing dimensions so that computing is lighter and suppresses overfitting.

**2.4.2. Fully Connected.** At this stage, transformation is carried out into one dimension. This aims to facilitate the application of the perceptron multilayer learning process. Figure 4 shows that the fully connected method is divided into three main activities: flatten, linear transformation, and nonlinear transformation.

**Flatten.** Flatten is a step to transform data into a matrix into one dimension matrix or array. It aims to conduct training data as well as the multilayer perceptron method. In this example, the illustration can be seen in Figure 6.

**Linear Transformation.** In applying feedforward, a linear regression technique provides a relationship between one layer and another. The linear transformation formula is shown in the following formula:

$$Z_2 = W^T \cdot X + b \quad (3)$$

where W is the random weight value, b is the bias value, and Z<sub>2</sub> is the result of linear transformation as shown in figure 4.

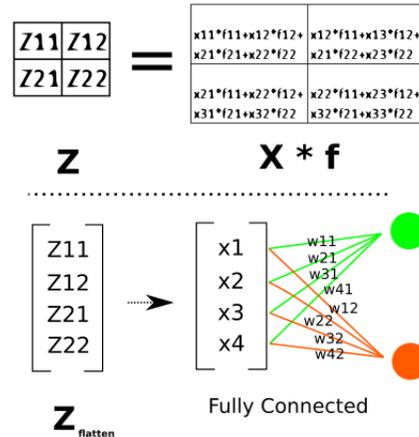


Fig. 6. Flatten

**2.4.3. Backpropagation.** Backpropagation is a method applied for learning from models that have been formed from feedforward. Backpropagation is a major part of every neural network model in terms of weight and filter improvements.

**Nonlinear Transformation.** Most cases that occur in the formation of models are nonlinear.

$$O = f(Z_2) \quad (4)$$

For this reason, the transformation of functions from linear to nonlinear can be done. Some transformation methods that can be applied are as shown in the following table:

Table 1. Nonlinear Transformation

Function	Formula
ReLU	$f(x) = \max(0, x)$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Softmax	$f(x) = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}}$

**Weight improvement.** The resulting error can be minimized by gradient descent, where the value derived from the value to be changed to zero. The mathematical notation can be written as follows:

$$\frac{\partial E}{\partial W} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial W} = 0 \quad (5)$$

Each formula and its derivatives can be seen in table below.

After knowing the value of , the improvement of the old value with the new one is:

**Table 2.** The Derivatives for Each Layer

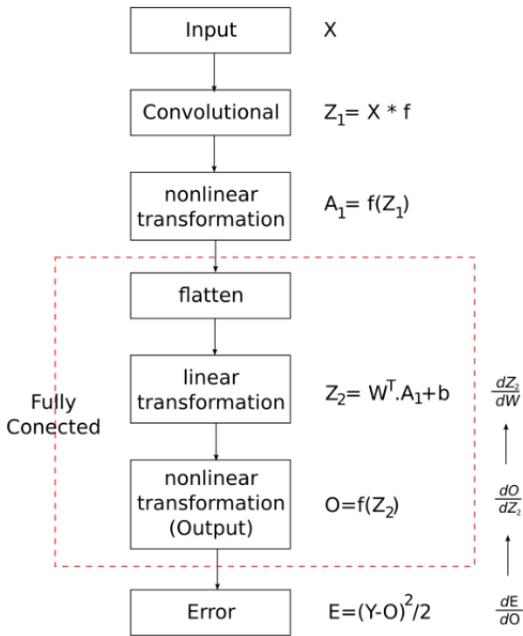
Formula	Derivative
$E = \frac{(Y-O)^2}{2}$	$\frac{\partial E}{\partial O} = -(Y - O)$
$O = f(Z_2)$	$\frac{\partial O}{\partial Z_2} = f'(Z_2)$
$Z_2 = W^T \cdot A + b$	$\frac{\partial Z_2}{\partial A} = W^T$
$A = f(Z_1)$	$\frac{\partial A}{\partial Z_1} = W^T$
$Z_1 = X \cdot f$	$\frac{\partial Z_1}{\partial f} = X$

$$\frac{\partial E}{\partial f} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial A} \cdot \frac{\partial A}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial f} = 0 \quad (7)$$

**Table 3.** The Derivatives for Each Layer

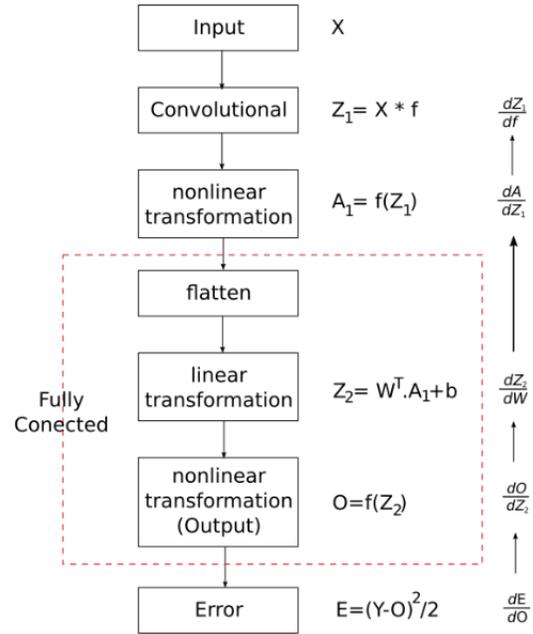
Formula	Derivative
$E = \frac{(Y-O)^2}{2}$	$\frac{\partial E}{\partial O} = -(Y - O)$
$O = f(Z_2)$	$\frac{\partial O}{\partial Z_2} = f'(Z_2)$
$Z_2 = W^T \cdot A + b$	$\frac{\partial Z_2}{\partial A} = W^T$

$$W_{new} = W_{old} - \frac{\partial E}{\partial W} \quad (6)$$



**Fig. 7.** backpropagation for weight improvement

**2.4.4. Making the image segmentation.** Model testing is carried out using the Monte Carlo cross-validation technique. The selection of training and testing data is made by taking samples without returning for training and testing.



**Fig. 8.** backpropagation for filter improvement

**Filter improvement.** Not only are weights needed to be improved, but filters also need to be updated in backpropagation activities. Following the same steps as in weight correction, the filter also needs to do the derivative of the error against the filter, and the value of this derivative is zero. The steps in question can be seen as in the figure 8. By using chain rules, it can be denoted as follows:

This iterative disagreement process, sometimes also called Monte Carlo Cross-Validation, provides a better estimate of how well the model can perform random training data compared to standard disagreement validation methods [13]. This was carried out several times and then seen the average and standard deviation of the resulting accuracy. The models that

will be included in the testing of this model are logistic regression, support vector machine, random forest, k-nearest neighbor, stochastic gradient descend, and naive-bayes. After that, compare the results of the six models.

### 3. Results and Explanation

There are three basic stages in implementing the proposed algorithm, namely selection and segmentation of images, modeling, and testing. Image selection and segmentation are used to estimate brain tumors' position in the image by giving black and white colors to a certain threshold. Modeling is used to make CNN models, while testing is done to determine the predicted images' accuracy with data outside of training data. The threshold selection results result by looking at the rules following table 1, visible images with a brain with a tumor and a normal brain. The difference is shown by the white grouping that appears in the figures in Table 2 and Table 3, where the brain that has a tumor has more color and tends to be asymmetrical. Some examples of the results of a threshold search experiment can be seen in the figure 9 and figure 10.

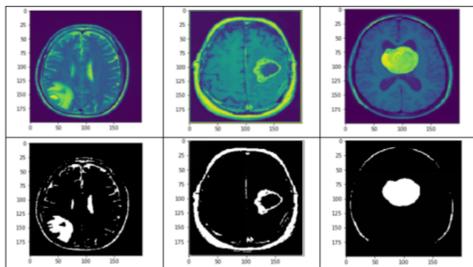


Fig. 9. The results of selection and segmentation for the brain affected by tumors

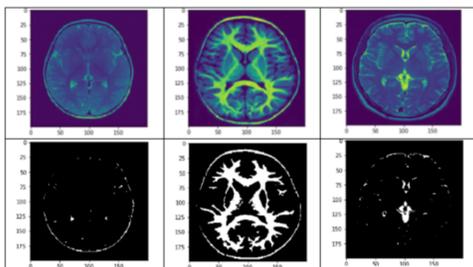


Fig. 10. The results of selection and segmentation for the health brain

Based on the pictures presented in tables 2 and 3, it appears that the automatic threshold selection based on the rules in table 1 can sort out contrasting

colors. This image is then collected into an array and then combined, then randomized the position to produce a model from the CNN method.

The model built has convolutional structures of 50 nodes, then fully connected with 100 nodes of relay layer, 100 nodes of sigmoid, and two softmax nodes. This model's final result is the chance of being selected as a tumor or normal at each node of the softmax results. This research will take the greatest opportunity as the final result of the decision.

Furthermore, a trial was conducted using the Monte Carlo method, where data was randomly randomized, then 25 percent was taken for testing, and the remainder was for training. Conducted as many as 100 experiments, calculated the level of accuracy, then averaged. The results of each model can be seen as in the following table:

Table 4. Accuracy for Each Classification Model

Model	Accuracy	Conf. Interval (95%)
Logistic Regression	70.81	$69.87 < X < 71.75$
SVM	68.45	$67.36 < X < 69.54$
Random Forest	73.69	$72.56 < X < 74.81$
kNN	52.93	$51.82 < X < 54.05$
SGD	71.64	$70.74 < X < 72.54$
Naive Bayes	69.46	$67.36 < X < 69.54$
CNN	75.42	$74.54 < X < 76.31$

The table above shows the average accuracy rate of 100 experiments conducted, with a 95 percent confidence interval. CNN can provide an average accuracy rate of 75.42 percent, with the accuracy interval being between 74.54 to 76.31 percent. This is a good sign compared to 6 other leading methods with an accuracy difference of 0.87 percent to Random Forest's accuracy.

Henceforth, it needs to be checked again whether the average accuracy between CNN and the random forest is significantly different. The steps taken are using the normality test in advance for 100 CNN accuracy data used and 100 random forest accuracy data. If these two data are normally distributed, then a t-test is performed to see whether the two results' average is significantly different. Normality test uses the Shapiro method with the following hypothesis:

$$\begin{aligned}
 H_0: & \text{Data distribution is normal} \\
 H_a: & \text{Data distribution is not normal} \\
 & \text{Reject } H_0 \text{ if } p_{value} < 0.05
 \end{aligned}$$

For CNN, the p-value of Shapiro test is 0.0077, which means Reject  $H_0$ . For Random Forest, the p-value is 0.0970, which means Accept  $H_0$ . the results shown by the convolutional model are not normally distributed, in contrast to the random

forest model. For this case, a Wilcoxon test was performed to check whether the two models produce significantly have different accuracy.

$$\begin{aligned} H_0 &: \bar{x}_{CNN} = \bar{x}_RF \\ H_a &: \bar{x}_{CNN} \neq \bar{x}_RF \\ \text{Reject } H_0 & \text{ if } p_{value} < 0.05 \end{aligned}$$

P-value obtained from existing data is 0.027, so it can be concluded that the accuracy of CNN is significantly different from Random Forest. Seeing the previous results that CNN is greater than Random Forest, then from the tests conducted, CNN gave significantly better results than other best models.

#### 4. Conclusion

Based on the results shown in table 4, it can be seen that CNN with the threshold selection method using histogram selection produces a model that can classify brain tumors with an accuracy of 75.42 percent and a confidence interval between 74.54 and 76.31 percent. This shows that by using this method, the algorithm can provide a fairly good level of accuracy compared with some other classification methods.

This study uses the CNN architecture as follows:

(1) convolutional as many as 50 nodes, then fully connected with (2) 100 nodes of relay layer, (3) sigmoid of 100 nodes, and (4) softmax of 2 nodes. However, it is possible to increase by considering the layer structure with different combinations and numbers of nodes for each layer.

#### References

- [1] E. N. Arrofiqoh and H. Harintaka, "Implementasi metode convolutional neural network untuk klasifikasi tanaman pada citra resolusi tinggi," *Geomatika*, vol. 24, no. 2, pp. 61–68, 2018.
- [2] F. F. Maulana and N. Rochmawati, "Klasifikasi citra buah menggunakan convolutional neural network," *Journal of Informatics and Computer Science*, vol. 1, no. 2, pp. 104–108, 2019.
- [3] J. Wang and Z. Li, "Research on face recognition based on cnn," in *IOP Conference Series: Earth and Environmental Science*, vol. 170, no. 3. IOP Publishing, 2018, p. 032110.
- [4] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deepest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.
- [5] S. G. Raja and K. Nirmala, "Detection of brain tumor using k- nearest neighbor (knn) based classification model and self-organizing map (som) algorithm," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, pp. 787–791, June 2019.
- [6] E. M. Husein and D. M. A. Mahmoud, "Brain tumor detection using artificial neural networks," *Journal of Science and Technology*, vol. 13, no. 2, 2012.
- [7] T. S. Kumar, K. Rashmi, S. Ramadoss, L. Sandhya, and T. Sangeetha, "Brain tumor detection using svm classifier," in *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*. IEEE, 2017, pp. 318–323.
- [8] P. Patel and M. Patel, "Tumor detection using particle swarm optimization to initialize fuzzy c-means," *International Journal for Scientific Research & Development*, vol. 4, no. 3, pp. 255–259, 2016.
- [9] K. Sudharani, T. Sarma, and K. S. Prasad, "Histogram related threshold techniques for region based automatic brain tumor detection," *Indian journal of science and technology*, vol. 9, p. 48, 2016.
- [10] C. Navoneel, "Data file (public-use): Brain mri images for brain tumor detection," Retrieved April 1st 2020 from <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>.
- [11] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, 2015.
- [12] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [13] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808*, 2018.