Available online at :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Accreditated SINTA "2" Kemenristek/BRIN, No. 85/M/KPT/2020

# Modification CNN Transfer Learning for Classification MRI Brain Tumor

**Retno Wardhani[1], Nur Nafi'iyah[2]**

[1.2] Departemen Teknik Informatika, Fakultas Teknik
Universitas Islam Lamongan, Lamongan, Indonesia
E-mail: retzno@yahoo.com[1], mynaff@unisla.ac.id[2]

**ARTICLE INFO**

***History of the article:***
Received March 20, 2023
Revised August 26, 2023
Accepted December 15, 2023

**Keywords:**
Modification layer
CNN transfer learning
MRI brain tumor

**Correspondece:**
E-mail: mynaff@unisla.ac.id

**ABSTRACT**

Identification, or detecting the infected part of a brain tumor on an MRI image, requires precision and takes a long time. MRI (Magnetic Resonance Imaging) is a magnetic resonance imaging technique to examine and take pictures of organs, tissues, and skeletal systems. The brain is essential because it is the center of the nervous system, which controls all human activities. Therefore, MRI of the brain has an important role, one of which is used for analysis or consideration before performing surgery. However, MRI images cannot provide optimal results when analyzed due to noise, and the bone and tumor (lumps of flesh) have the same appearance. AI (artificial intelligence), or digital image processing and computer vision, can analyze MRI images to detect or identify tumors correctly. This study proposes changes to the last layer of CNN (Convolution Neural Network) transfer learning (VGG16, InceptionV3, and ResNet-50) to identify brain tumor disease on MRI. Data were taken from Kaggle with types of glioma, meningioma, no tumor, and pituitary, with a total of 5712 training images and 1311 testing images. The proposed changes include a flattening layer and a pooling layer. The result is that replacing the flatten layer further improves accuracy, and the accuracy of the transfer learning CNNs (VGG16, InceptionV3, and ResNet-50) is 0.918, 0.762, and 0.934, respectively.

## INTRODUCTION

A brain tumor is a disease in which a lump of flesh grows in the brain (Sigit et al., 2019). The human brain is important because it is the body's nerve center. By utilizing MRI (Magnetic Resonance Imaging), imaging technology or magnetic resonance can be used to detect tumor disease in humans (Akbar et al., 2019). Can use MRI images of the brain for the initial stages in analyzing diseases of the brain as well as considerations in operating on the brain. Brain MRI images cannot diagnose the patient's disease precisely because there is noise or the same color appearance between the tumor tissue and other tissues. Several studies that have been conducted before mention that accuracy in diagnosing is crucial because it will affect the determination of the action given to the patient to cure the patient (Adinegoro et al., 2015; Soesanti et al., 2011; and Susmikanti, 2010). The stages in diagnosing or identifying diseases based on computer vision science include feature extraction or characteristics of image objects so that they can recognize patterns from an object and can be used for diagnosis. The feature extraction methods used in brain images in previous studies were tumor area, brain area, and presentation of tumor area to brain area, mean value, standard deviation, entropy, and variance of brain images (Akbar et al., 2019); Another feature used is LDA (Linear Discriminant Analysis) (Adinegoro et al., 2015); image mean value feature, image eigenbrain

(Soesanti et al., 2011); PCA (Principle Component Analysis) feature (Susmikanti, 2010); the feature used is the value GLCM (Gray Level Co-occurrence Matrix) (Widhiarso et al., 2018); another feature is the DWT (Discrete Wavelet Transformation) (Astuti, 2019), (Varuna Shree & Kumar, 2018), (Kumar et al., 2017); Histogram of Oriented Gradient (HOG) (M & Azizah, 2022); texture features with values of Contrast, Correlation, Energy, Dissimilarity, ASM (Angular Second Moment), Homogeneity, and Entropy (Febrianti et al., 2020).

The practice of using an image object's features to distinguish between things both similar and different is known as feature extraction. Shape, color, and texture traits can be used to differentiate between items in an image, according to research (Sari et al., 2014). In order to diagnose or differentiate between different types of sickness, it is important to extract features from visual objects. The diagnostic system will be able to discern between each type of disease or categorization with ease provided the object's traits or characteristics can accurately depict the object's overall image.

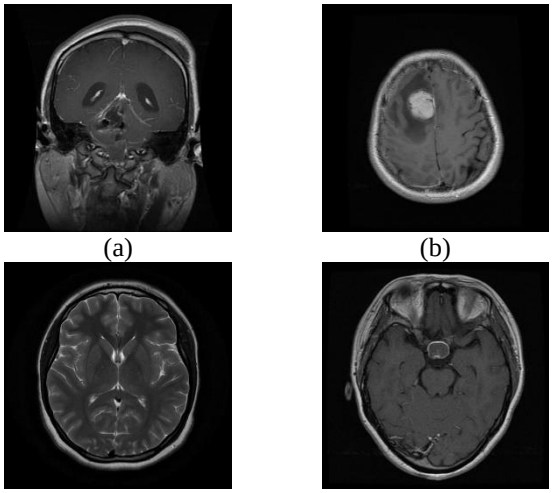
Recognizing an image object by its features is the process of diagnosing or recognizing it. While techniques based on machine learning or neural networks are employed to recognize or categorize variations or similarities in visual objects. Finding malignancies on brain MRI pictures or extracting features from the images has been the subject of numerous research. Several methods are used to identify or diagnose brain tumors Naive Bayes (Akbar et al., 2019); SVM (Adinegoro et al., 2015), (Kumar et al., 2017), (M & Azizah, 2022), (Febrianti et al., 2020); Multi-Layer Neural Network (Susmikanti, 2010), (Astuti, 2019), (Varuna Shree & Kumar, 2018), (Gu & Li, 2021); CNN (Convolution Neural Network) or Deep Learning (Widhiarso et al., 2018), (Tjahyaningtijas et al., 2021), (Irmak, 2021), (Aamir et al., 2022), (Harish & Baskar, 2020), (Abiwinanda et al., 2020), (Daniel & Ruxandra, 2021), (Deepak & Ameer, 2019), (Yuliawan & 'Uyun, 2022).

Many studies on MRI images have been carried out on the identification of tumor disease. However, some identification results must be optimal, especially if the image is unclear. We propose identifying the type of brain tumor disease on MRI images using the CNN method. From previous studies, many have used the neural network, or deep learning/CNN method. The results of several CNN methods are perfect, with an accuracy of 95% to 97% (Tjahyaningtijas et al., 2021), 92% to 99% (Irmak, 2021), 98% (Aamir et al., 2022), Daniel & Ruxandra, 2021, and Deepak & Ameer, 2019). Based on research (Irmak, 2021), we propose to use the CNN transfer learning method (VGG16, InceptionV3, and ResNet-50) and do layer changes at the end. This study aims to determine changes in accuracy from layer changes at the end of transfer learning (VGG16, InceptionV3, and ResNet-50).

## RESEARCH METHODS

### 1. Dataset

Data taken from Kaggle dataset (Kaggle, n.d.). Table 1 shows the type of tumor disease studied and the amount of image data used. The MRI image data used for training or evaluation is 256x256 in size, as shown in Figure 1.



(a) (b)

(c) (d)

Figure 1. (a) Glioma (b) Meningioma (c) No Tumor (d) Pituitary

## 2. Convolution Neural Network Transfer Learning

The research proposal is to change the layer at the end of the transfer learning (VGG16, InceptionV3, ResNet-50), as shown in Figure 2. The VGG16, InceptionV3, and ResNet-50 architectures are added by one layer, with alternative Flatten, Max Pooling, and Average layers pooling. At the end of the layer, we propose to add one layer, for example, VGG16 plus a Flatten layer, then be trained and evaluated. For example, VGG16 plus the Average Pooling layer was trained and evaluated. Each transfer learning (VGG16, InceptionV3, and ResNet-50) is added by one layer with three choices of layers, then trained and evaluated. The results of the evaluation process of the proposal, We calculate the accuracy as in Equation 1 (Varuna Shree & Kumar, 2018).

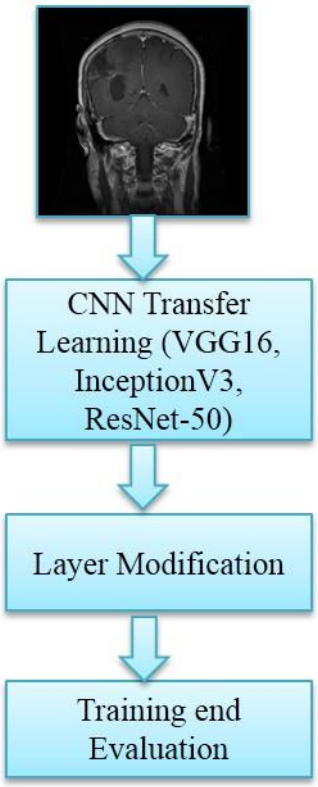



Figure 2. Research Propose

Table 1. Dataset

| *No* | *Type* | *Training* | *Testing* |
|---|---|---|---|
| 1 | Glioma | 1321 | 300 |
| 2 | Meningioma | 1339 | 306 |
| 3 | Notumor | 1595 | 405 |
| 4 | Pituitary | 1457 | 300 |
| | Total | 5712 | 1311 |

$$accuracy = \frac{\sum_{i=1} y_i = \hat{y}_i}{N} \quad (1)$$

## RESULTS AND DISCUSSION

We propose that VGG16 transfer learning be added one layer at the end; namely, Figure 3 the parameters of VGG16 if added to the Flatten layer, Figure 4 the parameters of VGG16 if added to the Average Pooling layer, Figure 5 the parameters of VGG16 when added to the Max Pooling layer. Figure 6, Figure 7, and Figure 8, respectively, show the process of changing the value of the loss function and accuracy during training from the addition of the Flatten layer, Average Pooling layer, and Max Pooling layer.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
vgg16 (Model)                (None, 8, 8, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 32768)             0
_________________________________________________________________
dense (Dense)                (None, 4)                 131076
=================================================================
Total params: 14,845,764
Trainable params: 131,076
Non-trainable params: 14,714,688
```

Figure 3. Adding flatten layer to VGG16

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
vgg16 (Model)                (None, 8, 8, 512)         14714688
_________________________________________________________________
global_average_pooling2d (Gl (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 4)                 2052
=================================================================
Total params: 14,716,740
Trainable params: 2,052
Non-trainable params: 14,714,688
```

Figure 4. Adding average pooling layer to VGG16

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
vgg16 (Model)                (None, 8, 8, 512)         14714688
_________________________________________________________________
global_max_pooling2d (Global (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 4)                 2052
=================================================================
Total params: 14,716,740
Trainable params: 2,052
Non-trainable params: 14,714,688
```

Figure 5. Adding max pooling layer to VGG16

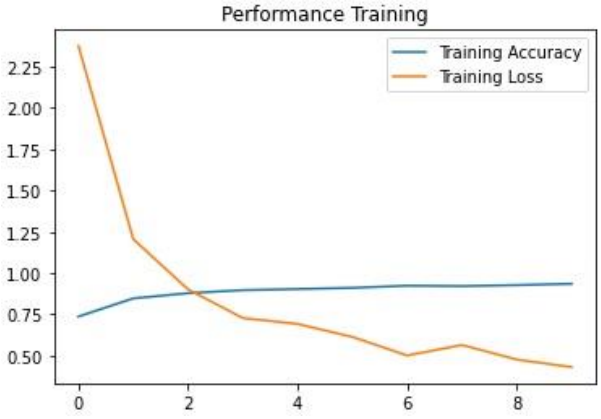



Figure 6. VGG16 training process

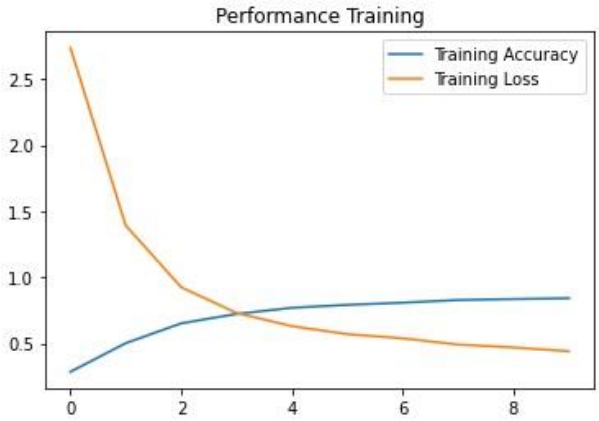



Figure 7. VGG16 training process

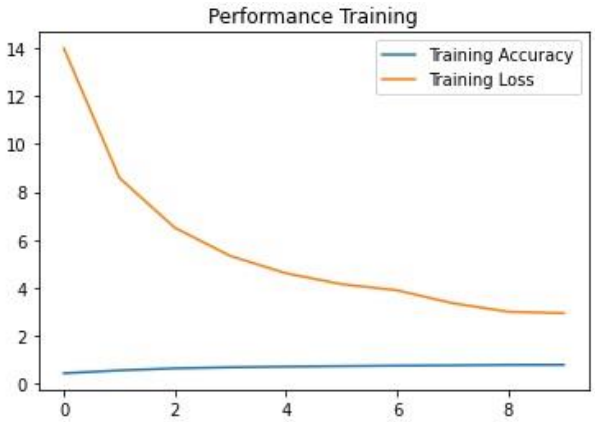



Figure 8. VGG16 training process

We propose that transfer learning InceptionV3 be added one layer at the end; namely, Figure 9 the parameters of InceptionV3 if added to the Flatten layer, Figure 10 the parameters of InceptionV3 if added to the Average Pooling layer, Figure 11 the parameters of InceptionV3 if added to the Max Pooling layer. Figure 12, Figure 13, and Figure 14, respectively, show the process of changing the value of the loss function and accuracy during training from the addition of the Flatten layer, Average Pooling layer, and Max Pooling layer.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_2 (InputLayer) | [(None, 256, 256, 3)] | 0 |
| inception_v3 (Model) | (None, 6, 6, 2048) | 21802784 |
| flatten (Flatten) | (None, 73728) | 0 |
| dense (Dense) | (None, 4) | 294916 |

Total params: 22,097,700
Trainable params: 294,916
Non-trainable params: 21,802,784

Figure 9. Adding flatten layer to InceptionV3

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_2 (InputLayer) | [(None, 256, 256, 3)] | 0 |
| inception_v3 (Model) | (None, 6, 6, 2048) | 21802784 |
| global_average_pooling2d (Gl | (None, 2048) | 0 |
| dense (Dense) | (None, 4) | 8196 |

Total params: 21,810,980
Trainable params: 8,196
Non-trainable params: 21,802,784

Figure 10. Adding average pooling layer to InceptionV3

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_2 (InputLayer) | [(None, 256, 256, 3)] | 0 |
| inception_v3 (Model) | (None, 6, 6, 2048) | 21802784 |
| global_max_pooling2d (Global | (None, 2048) | 0 |
| dense (Dense) | (None, 4) | 8196 |

Total params: 21,810,980
Trainable params: 8,196
Non-trainable params: 21,802,784

Figure 11. Adding max pooling layer to InceptionV3

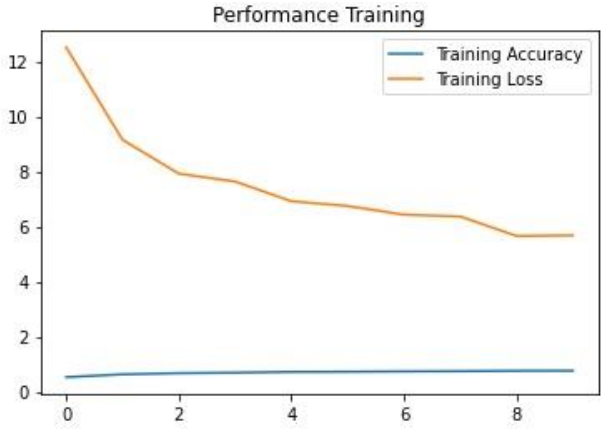



Figure 12. InceptionV3 training process

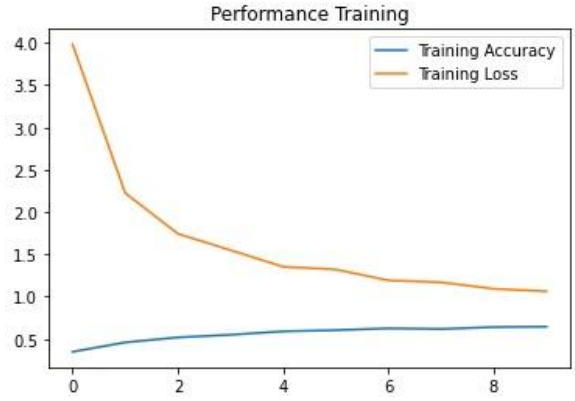



Figure 13. InceptionV3 training process

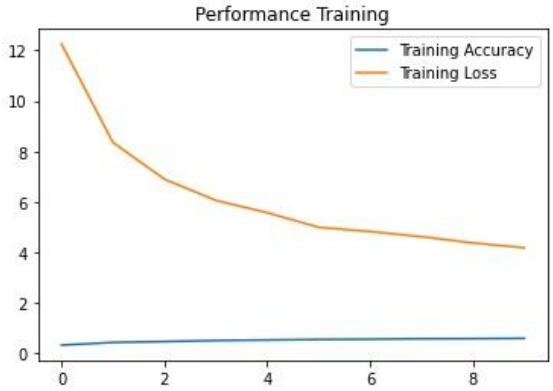



Figure 14. InceptionV3 training process

We propose a transfer learning ResNet-50 be added one layer at the end, namely Figure 15 the parameters of ResNet-50 if added to the Flatten layer, Figure 16 the parameters of ResNet-50 if added to the Average Pooling layer, Figure 17 the of parameters ResNet-50 if Max Pooling layer is added. Figure 18, Figure 19, and Figure 20, respectively, show the process of changing the value of the loss function and accuracy during training from the addition of the Flatten layer, Average Pooling layer, and Max Pooling layer.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
resnet50 (Model)             (None, 8, 8, 2048)        23587712
_________________________________________________________________
flatten (Flatten)            (None, 131072)            0
_________________________________________________________________
dense (Dense)                (None, 4)                 524292
=================================================================
Total params: 24,112,004
Trainable params: 524,292
Non-trainable params: 23,587,712
```

Figure 15. Adding flatten layer to ResNet-50

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
resnet50 (Model)             (None, 8, 8, 2048)        23587712
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 4)                 8196
=================================================================
Total params: 23,595,908
Trainable params: 8,196
Non-trainable params: 23,587,712
```

Figure 16. Adding average pooling layer to ResNet-50

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
resnet50 (Model)             (None, 8, 8, 2048)        23587712
_________________________________________________________________
global_max_pooling2d (Global (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 4)                 8196
=================================================================
Total params: 23,595,908
Trainable params: 8,196
Non-trainable params: 23,587,712
```

Figure 17. Adding max pooling layer to ResNet-50

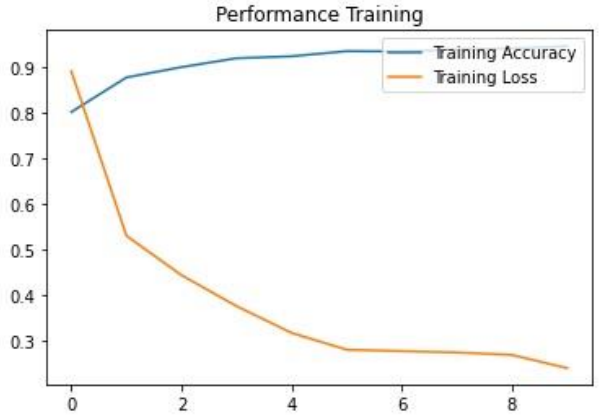



Figure 18. ResNet-50 training process


Figure 19. ResNet-50 training process

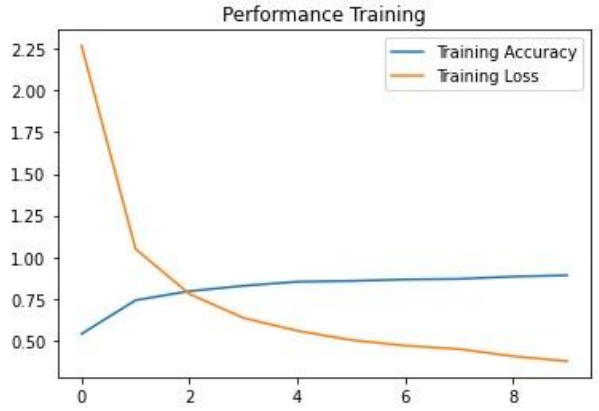



Figure 20. ResNet-50 training process

Figures 3 to 20 are the total parameters of each CNN transfer learning model (VGG16, InceptionV3, ResNet-50) and model the transfer learning training process, which is replaced by the last layer of each transfer learning model. Each transfer learning model replaced by the last layer is trained on ten epochs of MRI data, and each epoch reads 200 images. The training process used the rmsprop optimizer with a learning rate of 0.0001 and loss function categorical cross-entropy. The model obtained from the training was tested on the data and evaluated for accuracy. Experiments identify test data from all transfer learning models, replaced by the last layer in Table 2. Each accuracy value of the added layer on Flatten, Average Pooling, and Max Pooling is shown in Table 2. Based on Table 2, adding a flattening layer has the highest accuracy compared to adding other layers. The transfer learning model with the highest accuracy is ResNet-50.

Table 2. Accuracy Results

| *CNN* | *Flatten* | *Average Pooling* | *Max Pooling* |
|---|---|---|---|
| VGG16 | 0.918 | 0.802 | 0.852 |
| InceptionV3 | 0.762 | 0.602 | 0.592 |
| ResNet-50 | 0.934 | 0.871 | 0.918 |

## CONCLUSIONS AND RECOMMENDATIONS

We classified tumor disease on MRI images using the CNN transfer learning method (VGG16, InceptionV3, and ResNet-50). Proposal We add one layer at the end for each VGG16, InceptionV3, and ResNet-50. We made three selections, adding one layer at the end: the flatten layer, the average pooling layer, and the max pooling layer. By adding a flattening layer, the highest accuracy results are 0.934 on ResNet-50, 0.918 on VGG-16, and 0.762 on InceptionV3. The CNN transfer learning with the highest accuracy is ResNet-50. Suggestions for further research to build other transfer learning models and improve the architecture to improve accuracy.

## REFERENCES

Aamir, M., Rahman, Z., Dayo, Z. A., Abro, W. A., Uddin, M. I., Khan, I., Imran, A. S., Ali, Z., Ishfaq, M., Guan, Y., & Hu, Z. (2022). A deep learning approach for brain tumor classification using MRI images. *Computers and Electrical Engineering*, *101*(May), 108105. https://doi.org/10.1016/j.compeleceng.2022.108105

Abiwinanda, N., Hanif, M., Hesaputra, S. T., Handayani, A., & Mengko, T. R. (2020). Brain tumor classification using convolutional neural network. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11993 LNCS*, 335–342. https://doi.org/10.1007/978-3-030-46643-5_33

Adinegoro, A., Atmaja, R. D., & Purnamasari, R. (2015). Deteksi Tumor Otak dengan Ektrasi Ciri & Feature Selection mengunakan Linear Discriminant Analysis (LDA) dan Support Vector Machine (SVM). *E-Proceeding of Engineering*, *2*(2), 2532–2539.

Akbar, F., Rais, A. N., Sobari, I. A., Zuama, R. A., & Rudiarto, B. (2019). Analisis Performa Algoritma Naive Bayes pada Deteksi Otomatis Citra MRI. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *5*(1). https://doi.org/10.33480/jitk.v5i1.586

Astuti, L. W. (2019). Ekstrasi Fitur Citra MRI Otak Menggunakan Data Wavelet Transform (DWT) untuk Klasifikasi Penyakit Tumor Otak. *Jurnal Ilmiah Informatika Global*, *10*(2), 80–86. https://doi.org/10.36982/jig.v10i2.854

Daniel, M. C., & Ruxandra, L. M. (2021). Brain Tumor Classification Using Pretrained Convolutional Neural Networks. *2021 16th International Conference on Engineering of Modern Electric Systems, EMES 2021 - Proceedings*, *11*(September), 1457–1461. https://doi.org/10.1109/EMES52337.2021.9484102

Deepak, S., & Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. *Computers in Biology and Medicine*, *111*(June), 103345.

https://doi.org/10.1016/j.compbiomed.2019.103345

Febrianti, A. S., Sardjono, T. A., & Babgei, A. F. (2020). Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vector Machine. *Jurnal Teknik ITS*, *9*(1). https://doi.org/10.12962/j23373539.v9i1.51587

Gu, Y., & Li, K. (2021). A Transfer Model Based on Supervised Multi-Layer Dictionary Learning for Brain Tumor MRI Image Recognition. *Frontiers in Neuroscience*, *15*. https://doi.org/10.3389/fnins.2021.687496

Harish, P., & Baskar, S. (2020). MRI based detection and classification of brain tumor using enhanced faster R-CNN and Alex Net model. *Materials Today: Proceedings*, *xxxx*. https://doi.org/10.1016/j.matpr.2020.11.495

Irmak, E. (2021). Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework. *Iranian Journal of Science and Technology - Transactions of Electrical Engineering*, *45*(3). https://doi.org/10.1007/s40998-021-00426-9

Kaggle. (n.d.). *Dataset MRI*. https://www.kaggle.com/masoudnickparvar/brain-tumor-mri-dataset

Kumar, S., Dabas, C., & Godara, S. (2017). Classification of Brain MRI Tumor Images: A Hybrid Approach. *Procedia Computer Science*, *122*, 510–517. https://doi.org/10.1016/j.procs.2017.11.400

M, T. A., & Azizah, Q. N. (2022). *Klasifikasi Tumor Otak Menggunakan Ekstraksi Fitur HOG dan Support Vector Machine*. *4*(1), 45–50.

Sari, Y. A., Dewi, R. K., & Fatichah, C. (2014). Seleksi Fitur Menggunakan Ekstraksi Fitur Bentuk, Warna, dan Tekstur dalam Sistem Temu Kembali Citra Daun. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *12*(1). https://doi.org/10.12962/j24068535.v12i1.a39

Sigit, R., Wulandari, A., Rofiqah, N., & Yuniarti, H. (2019). Automatic detection brain segmentation to detect brain tumor using MRI. *International Journal on Advanced Science, Engineering and Information Technology*, *9*(6). https://doi.org/10.18517/ijaseit.9.6.8536

Soesanti, I., Susanto, A., Widodo, T., & Tjokronagoro, M. (2011). Ekstraksi Ciri dan Identifikasi Citra Otak MRI Berbasis Eigenbrain Image. *Forum Teknik*, *34*(1).

Susmikanti, M. (2010). *Pengenalan Pola Berbasis Jaringan Syaraf Tiruan Dalam Analisa CT Scan Tumor Otak Beligna*. *2010*(Snati), 26–31.

Tjahyaningtijas, H. P. A., Rumala, D. J., Angkoso, C. V., Fanani, N. Z., Santoso, J., Sensusiati, A. D., Ooijen, P. M. A. V., Ketut Eddy Purnama, I. K. E., & Purnomo, M. H. (2021). Brain Tumor Classification in MRI Images Using En-CNN. *International Journal of Intelligent Engineering and Systems*, *14*(4). https://doi.org/10.22266/ijies2021.0831.38

Varuna Shree, N., & Kumar, T. N. R. (2018). Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network. *Brain Informatics*, *5*(1). https://doi.org/10.1007/s40708-017-0075-5

Widhiarso, W., Yohannes, Y., & Prakarsah, C. (2018). Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, *8*(2), 179. https://doi.org/10.22146/ijeis.34713

Yuliawan, E., & 'Uyun, S. (2022). Chest X-ray Image Classification for COVID-19 diagnoses.pdf. *JISEBI*, *8*(2), 109–118.