Brain tumor detection

Tin Popović
Fakultet elektrotehnike i računarstva
Zagreb, Croatia
JMBAG: 0036526279
tin.popovic@fer.hr

Mislav Prce
Fakultet elektrotehnike i računarstva
Zagreb, Croatia
JMBAG: 0036518761
mislav.prce@fer.hr

Oleksii Chumakov
Fakultet elektrotehnike i računarstva
Zagreb, Croatia
JMBAG: 0036554627
oleksii.chumakov@fer.hr

David Konjevod
Fakultet elektrotehnike i računarstva
Zagreb, Croatia
JMBAG: 0036518805
david.konjevod@fer.hr

Ivan Joskić
Fakultet elektrotehnike i računarstva
Zagreb, Croatia
JMBAG: 0036517353
ivan.joskic@fer.hr

Matko Barbić
Fakultet elektrotehnike i računarstva
Zagreb, Croatia
JMBAG: 0036525698
matko.barbic@fer.hr

***Abstract*— The International Agency for Research on Cancer (IARC) estimates that one in five people will develop cancer during their lifetime. The increase in mortality from cancer is attributed to the general aging of the population. In 2020, according to the World Health Organization (WHO), the cause of death of about 10 million people in the world was cancer. One type of cancer is brain tumor. The chances of survival in cancer patients depend on the stage at which the cancer is detected: the sooner the better. Therefore, detecting a brain tumor as early as possible in magnetic resonance imaging (MRI) scans is a critical topic. This paper presents a method to predicting brain tumors by using convolution neural networks (CNNs), where at the beginning, MRI images were segmented and cropped. Kaggle brain MRI dataset is used to train and evaluate the model to get maximized accuracy.**

Keywords*— *MRI Image, image segmentation, medical images processing, brain tumor detection, convolutional neural network

I INTRODUCTION

A brain tumor is all neoplasms of a benign or malignant nature that develop in the head organ itself (primary) or as a result of metastasis of cancer that has affected another organ (secondary). Malignant ones are distinguished by infiltrative growth (inside the surrounding tissues), the absence of clear boundaries, and the presence of metastases. They are dangerous to human life. While benign neoplasms are enclosed in a capsule. They do not metastasize and do not have infiltrative growth. Such neoplasms can be removed and they do not recur. Benign neoplasms of the central nervous system can be no less dangerous than malignant ones. Although they grow slowly and respond well to treatment, in an advanced stage they can cause disability or death of the patient.

Diagnosis, due to the localization of the tumor inside the skull, is difficult. MRI is one of the main methods for diagnosing and differentiating tumors of the brain and spinal cord. MRI in different projections allows you to obtain diagnostic information about the location, size and structure of the tumor. According to the results of magnetic resonance imaging, doctors can detect changes in the structure of tissues. Doctors suspect a pathological formation by the type and uniformity of changes in the MR signal. One of the most difficult stages is segmentation (feature extraction) for classification into a malignant or benign tumor.

Artificial neural networks (ANNs) expand clinical possibilities in the diagnosis and therapy of malignant and benign neoplasms. ANNs helps make cancer care more personalized. Differential diagnosis of a tumor, probabilistic assessment of outcomes, prediction of response to therapy are opportunities for an individual approach to the patient. During processing, image segmentation is carried out with the identification of suspicious zones. First, the possible malignancy of the tumor in the selected area is determined. Then it is assigned a stage in accordance with generally accepted scales.

Further, convolutional neural networks (CNNs) will be used. They are one of the most used ANNs that work through a convolution process.

II LITERATURE REVIEW

The most well-known image identification and classification algorithm is a convolutional neural network (CNN).
One of the important developments that sparked the deep neural network renaissance in computer vision, a subset of machine learning, was the development of CNNs. Typically, convolutional, pooling, and dense layers are combined to create a CNN.

Convolutional networks are multilayer perceptrons with a high degree of invariance to translation, scaling, skewing and other types of distortion. They are specifically designed to recognize two-dimensional shapes. Through the use of a network with the following types of constraints built into its structure, this challenging task is learned in a supervised manner:

1. *Feature extraction:* Each neuron receives synaptic inputs from a particular local receptive field in the layer before, which forces it to identify local features.
2. *Feature mapping:* Each of the network's computational layers is made up of several feature maps, each of which is a plane where each individual neuron shares the same set of synaptic weights. The benefits of this approach are *shift invariance* and *reduction in the number of free parameters*.
3. *Subsampling:* The resolution of the feature map is decreased by a computational layer that follows each convolutional layer and carries out local averaging and subsampling. The feature map's outcome is less sensitive to shifts and other types of distortion due to this operation.

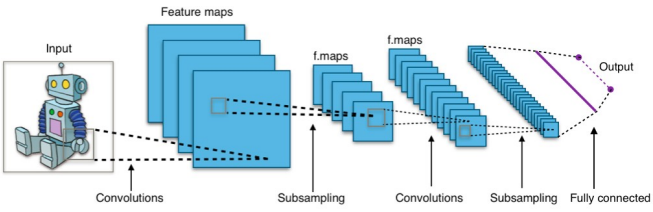

Figure depicts the convolutional network's architecture, which consists of an input layer, four hidden layers and an output layer. This network was created to process images (e.g., recognition of handwritten characters).
The input layer, which is made up of sensory nodes, is where the various character pictures that have been centered and normalized in size are received. Then, convolution and subsampling are alternated in the computational layouts.

The number of feature maps increases while the spatial resolution decreases as the computational layers advance, each of which alternates between convolution and sub-sampling. Convolution followed by subsampling is a concept with neurobiological roots (simple cells followed by more complex cells).

Through weight sharing, the number of free parameters is drastically reduced. As a result, the learning machine's capacity is reduced, which enhances its generalization ability. Another important aspect is that the convolutional network can be implemented in parallel form thanks to the usage of weight sharing.

III METHOD

In this paper, two CNN models are implemented and then used as classification algorithms for brain tumor detection. Both models use a Kaggle dataset that has 253 images, where 155 of them have tumors and 98 do not have tumors.

III.A First CNN model

The first model's starting point is the dataset we introduced previously. The Kaggle data set has two problems. It is unbalanced and only has 253 images. The size of our data set is not large enough to train the network that we are trying to implement. That is why we will increase the size of our dataset using data augmentation. In this paper each image with a tumor is a tumor is segmented into 6 images, and an image with no tumors is segmented into 9 images. New images where generated using *ImageDataGenerator* from *Keras* library. With this data augmentation technique, we got a data set of 2065 images where 1085 samples contain tumors (53%), and 980 samples do not (47%). The process of data augmentation lasted almost 2 minutes.
After we had successfully generated new data, we need to take a closer look at our images. We know that we have MRI images and that the only important part of our picture is the brain. The black cover behind the brain doesn't have any information, so we must get rid of it. Image cropping was done with *cv2* library. The image was first converted to a gray scale and blurred. Then the threshold, erode, dilate, findContours functions from the *cv2* package were used to find the extreme point of the MRI image. In the end, the brain was cropped from the image. The difference between the original and cropped image can be seen on Figure III-1 Original vs cropped image.

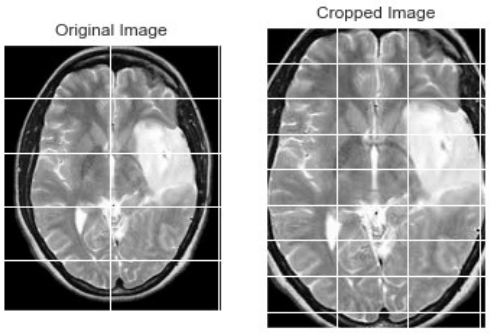

Figure III-1 Original vs cropped image

After cropping all the images, we need to resize all the photos to have the same width, shape, and number of channels. In this project, all pictures have a format of (240,240,3). After resizing, we need to apply normalization so that the pixel values are in the range 0-1. In Figure III-2 Images from dataset, we can see what our images look like.

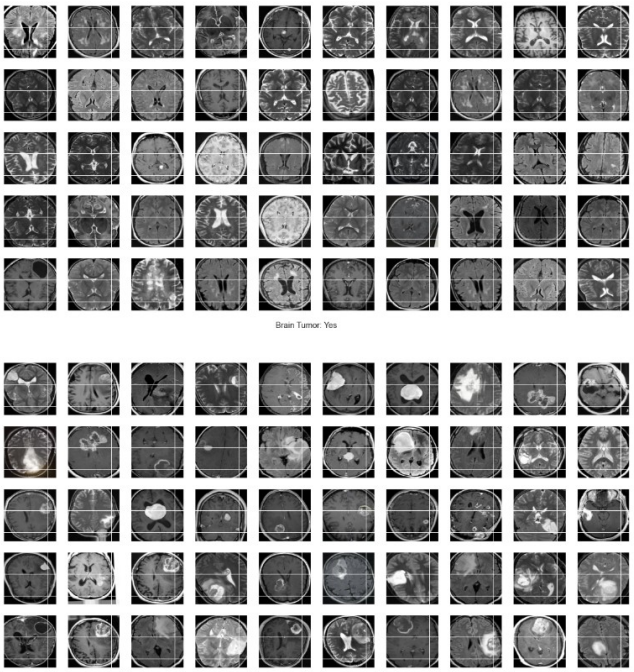

Figure III-2 Images from dataset

After data argumentation and data processing, the next step is implementing the model. The structure of the first model that we implemented is displayed on Figure III-3 Structure of CNN.

Figure III-3 Structure of CNN

The second layer in our network is a ZeroPadding2D layer. This layer can add rows and columns of zeros at the top, bottom, left and right sides of an image tensor. The next layer is a 2D Convolution Layer, a layer that creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. The next layer is the batch normalization layer which applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The ReLu function is used as the activation function. Next, we have two MaxPooling2D layers. Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map

```
Model: "BrainDetectionModel"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 240, 240, 3)]     0

 zero_padding2d (ZeroPadding  (None, 244, 244, 3)      0
 2D)

 conv0 (Conv2D)              (None, 238, 238, 32)      4736

 bn0 (BatchNormalization)    (None, 238, 238, 32)      128

 activation (Activation)     (None, 238, 238, 32)      0

 max_pool0 (MaxPooling2D)    (None, 59, 59, 32)        0

 max_pool1 (MaxPooling2D)    (None, 14, 14, 32)        0

 flatten_2 (Flatten)         (None, 6272)              0

 fc (Dense)                  (None, 1)                 6273

=================================================================
Total params: 11,137
Trainable params: 11,073
Non-trainable params: 64
_________________________________________________________________
```

containing the most prominent features of the previous feature map. Then we have a layer that flattens the input and does not affect the batch size. At the end we have a dense layer, the most common layer and frequently used layer. The dense layer returns the output of the operation: output=activation(dot(input, kernel) + bias).

III.B Second CNN model

The same Kaggle dataset was used for this second CNN model. However, the size of the original dataset was increased to 310 so that there is a balance between the two classes. The increase of the dataset was done by upscaling method. Before building the model the data was augmented. The data went through a series of changes including rescaling, random flipping, random rotation and random zooming.
This CNN model contains the following layers: input, GlobalAveragePooling2D, Batch Normalization, Dropout and Dense layer. We have described some of the layers that are also used in the first CNN model. Here we also have GlobalAveragePooling2D layer that applies global average pooling operation for spatial data, similar to MaxPooling2D. The dense layer uses the sigmoid as an activation function. The optimizer used in this paper is Adam with a learning rate of 1e-4.

IV RESULTS.

IV.A Results of the first CNN model

The dataset was split into training, testing and validation set. Train set contains 70% of data, and test and validation set both contain 15% of data. The model was trained with 20 epochs and with batch size of 32. The training process lasted about 25 minutes and gave a model with validation accuracy 0.89 and training accuracy 0.95. *Figure IV-1 Loss and accuracy change* shows the change of accuracy and loss in each iteration on the train and validation set.

Figure IV-4 Loss and accuracy change

The model with the highest validation accuracy (0.89) was used for testing the model on the test dataset. The testing process ended with accuracy of 0.88

IV.B Results of the second model

The model was trained on a subset of the dataset which has a 80-20 train-test split which means 248 train images and 62 test images. The model was trained for over 50 epochs and developed an accuracy of 96% on the validation dataset. In Figure IV-6 Tumor heatmap, the heatmap of the model is displayed with the warmer colors corresponding to the tumor and colder colors corresponding to normal brain tissue. The heatmap was developed based on data from the penultimate layer of the network so the images shown on the figure are not exactly the same as the final result of the network.

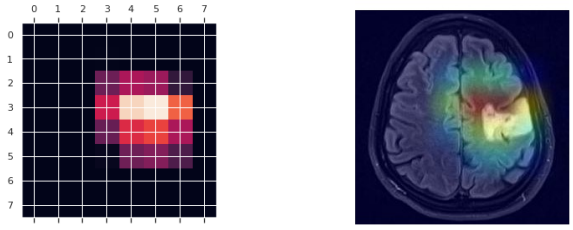

Figure IV-6 Tumor heatmap

In Figure IV-5 Confusion matrix we can see the results of the predictions of the model where a tumor presence or prediction of said tumor presence is marked with 1 and no tumor presence is marked with 0.

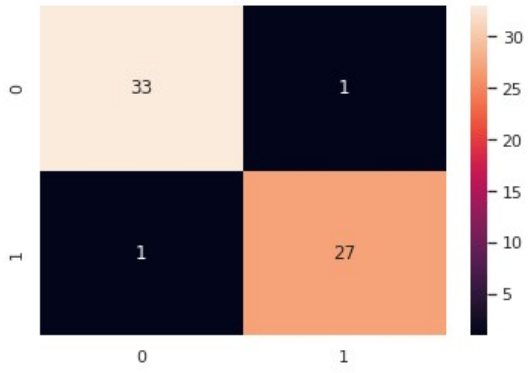

Figure IV-5 Confusion matrix

V Conclusion and discussion

Clinical therapy for brain tumors depend on accurate diagnosis. Medical images vary widely, thus it is essential to interpret them. Though easier to detect, the automatic brain tumor detection method considerably improves the patient's odds of surviving. Convolutional networks for classifying brain tumors have paved the path for more accurate and reliable tumor identification. Brain malignSancies are most frequently detected and categorized using MRI. When compared to traditional classification algorithms for medical imaging, DL-based techniques have recently attracted more attention and efficiency due to their apparent efficient feature extraction capacity. Using quick and inexpensive diagnostic technologies, the right grade of cancer can be identified, potentially saving many lives. Therefore, the development of quick, non-invasive, and affordable diagnostic methods is crucial. This study makes an effort to address these concerns by presenting a unique method for more accurate brain tumor detection in MRI images. For a remarkable accuracy rate, the suggested model combined deep learning and transfer learning techniques. In this study, the optimization of training models was boosted in comparison to similar strategies from earlier literature to lessen the demand for high processing power.

The designed and trained model was evaluated using cross-validation. The dataset was split between training, testing and validation data with the ratio 70:15:15, respectively. Using this evaluation method, the model

achieved an accuracy score of 90% on the validation dataset.

When compared to other models on Kaggle, we can say that our model has attained a decent degree of accuracy.

There is constant room for development. Depending on the situation, new models have been devised that deliver results that are either more precise or quicker working. Ours is adequate as a general analysis of CNN models.

REFERENCES

1 *S. Haykin, Neural Networks - A Comprehensive Foundation, Second Edition, 1999.*

2 *AI Wiki, CNN, https://machine-learning.paperspace.com/wiki/convolutional-neural-network-cnn*