Introduction to Artificial Intelligence and Machine Learning

PROJECT REPORT

Brain Tumor Identification And Classification Using Machine Learning

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, NITW

Submitted by:
Potnuri Sri Anjali Pravallika (21ECB0A45)

Supervisor:

Sri. K. Ravi Kishore Sir

Objective :

This project is “Brain Tumor Detection” and is made by using the KNIME Analytic Platform as the base for creating this helpful project. It is made by using Machine Learning Algorithms like Convolutional Neural Networks etc. Mainly it’s successful by using Transfer Learning.

Problem Statement:

A Brain tumor is considered as one of the most aggressive diseases, among children and adults. Brain tumors account for 85 to 90 percent of all primary Central Nervous System(CNS) tumors. Every year, around 11,700 people are diagnosed with brain tumors. The 5-year survival rate for people with a cancerous brain or CNS tumor is approximately 34 percent for men and 36 percent for women.

Brain Tumors are classified as Benign Tumors, Malignant Tumors, Pituitary tumors, etc. Proper treatment, planning, and accurate diagnostics should be implemented to improve the life expectancy of the patients. The best technique to detect brain tumors is Magnetic Resonance Imaging (MRI). A huge amount of image data is generated through the scans. These images are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties.

Application of automated classification techniques using Machine Learning(ML) and Artificial Intelligence(AI)has consistently shown higher accuracy than manual classification. Hence, proposing a system performing detection and classification by using Deep Learning Algorithms using convolution Neural Networks (CNN), Artificial Neural Networks (ANN), and transfer learning (TL) would be helpful to doctors all around the world.

Data Collection:

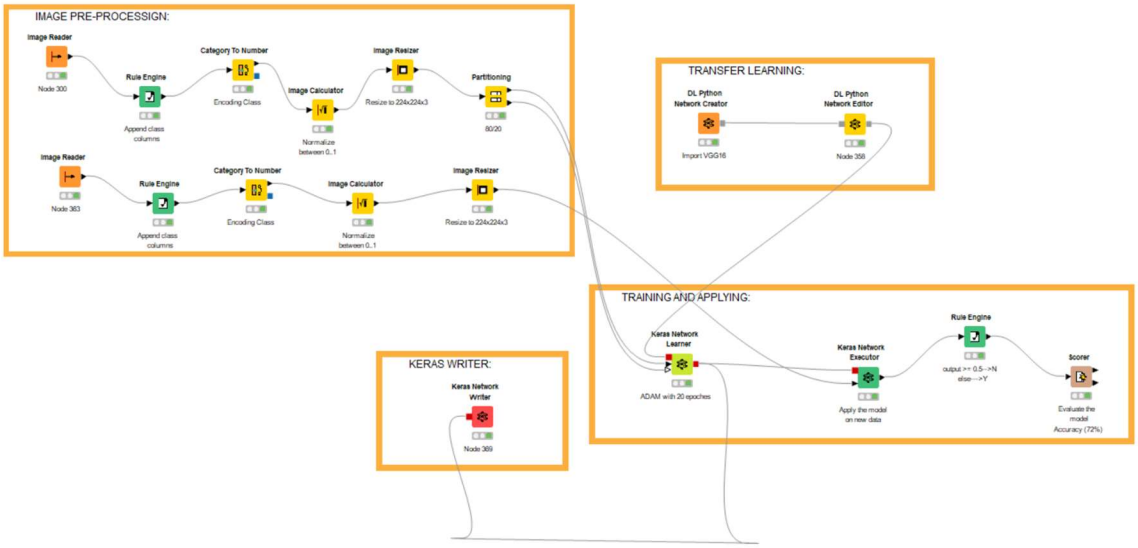
Brain Tumors are complex. There are many abnormalities in the size and location of the brain tumor(s). This makes it really difficult to completely understand the nature of the tumor. Also, a professional Neurosurgeon is required for MRI analysis. Often times in developing countries the lack of skillful doctors and lack of knowledge about tumors makes it really challenging and time-consuming to generate reports from MRIs. So in this project, it's an automated system that lets us know if the person has tumors.

I collected data from Kaggle, the below are the references of websites:

https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

https://www.kaggle.com/code/loaiabdalslam/brain-tumor-mri-classification-vgg16/data

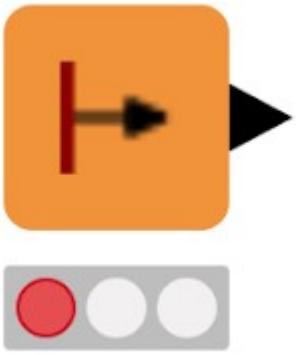
Procedure:



The above is the picture of the project workflow(Training)

In the workflow, I included nodes namely,

Image Reader:

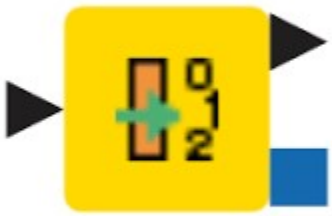


Reads images from various file formats supported by the Bio-Formats library and imports them to KNIME-internal image format.

Rule Engine:

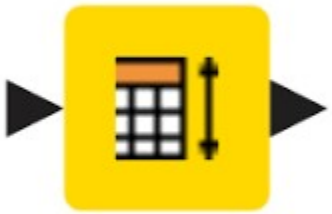
Takes a list of user-defined rules and tries to match them to each row in the input. If it matches, the outcome value is added in a new column.

Category to Number:



Takes columns with nominal data and maps every category to an integer.

Image Calculator:

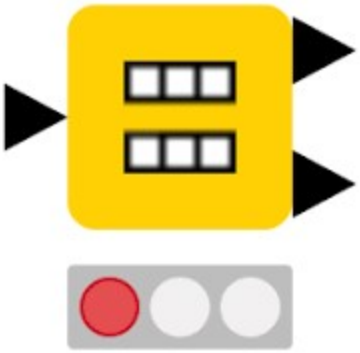


Evaluates(a free-form) mathematical expression based on images in a row.

Image Resizer:

Resizes the image in each dimension. The resulting values at each position in the image are set according to the resizing strategy.

Partitioning:



The input table is split into two partitions, train, and test data.

DL Python Network Creator:

It allows the custom creation of a Python-compatible deep learning network in a local Python installation via a user-defined script.

DL Python Network Editor:

Allows custom editing of a Python-compatible deep learning network in a local Python installation via a user-defined script.

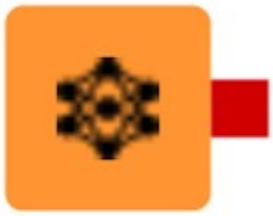
Keras Network Learner:

Performs supervised learning on a Keras deep learning network.

Keras Network Executor:

Executes a Keras deep learning network on a compatible external back end that can be selected by the user.

Keras Network Reader:

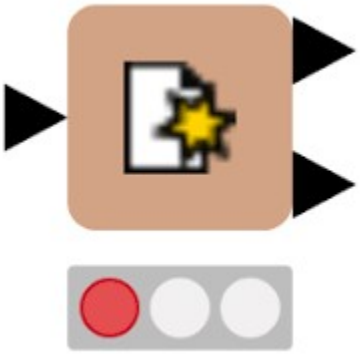


Reads a Keras deep learning network from an input file, the file can either contain a full, pre-trained network(.h5 file) or just a network specification without weights (.json or .yaml file).

Keras Network Writer:

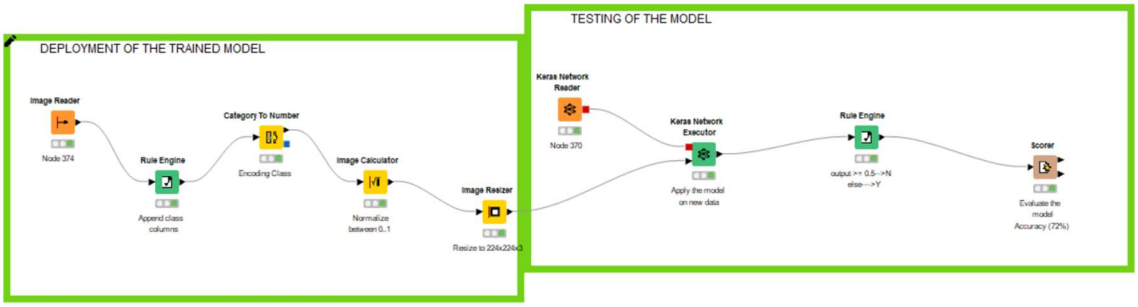
Writes a Keras network to a file.

Scorer:



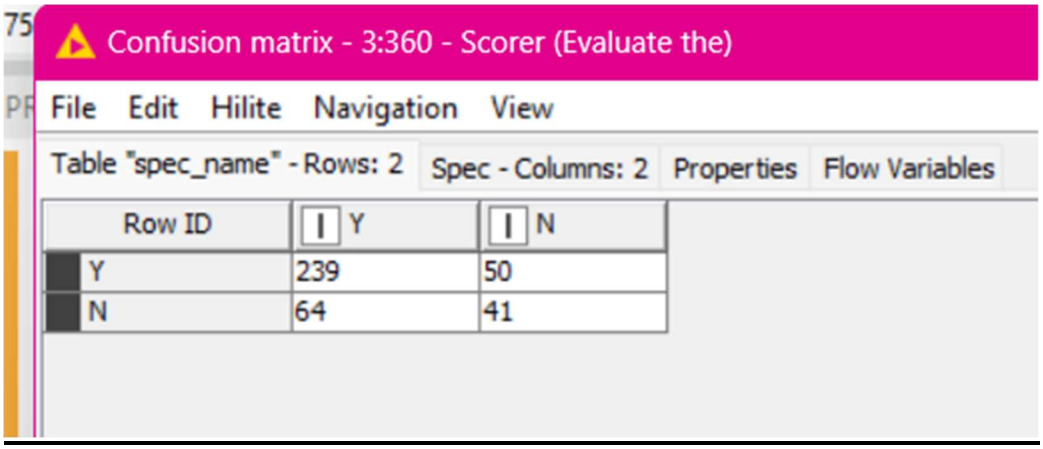
Compares two columns by their attribute value pairs and shows the confusion matrix, how many rows of which attribute and their classification match.

The deployment of the model:



Results:

Confusion Matrix:

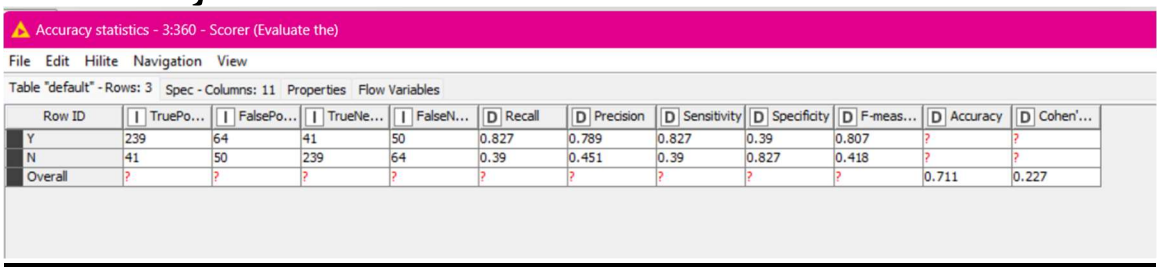

Confusion matrix - 3:360 - Scorer (Evaluate the)

File Edit Hilite Navigation View

Table "spec_name" - Rows: 2 | Spec - Columns: 2 | Properties | Flow Variables

Row ID	Y	N
Y	239	50
N	64	41

Accuracy Statistics:


Accuracy statistics - 3:360 - Scorer (Evaluate the)

File Edit Hilite Navigation View

Table "default" - Rows: 3 | Spec - Columns: 11 | Properties | Flow Variables

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
Y	239	64	41	50	0.827	0.789	0.827	0.39	0.807	?	?
N	41	50	239	64	0.39	0.451	0.39	0.827	0.418	?	?
Overall	?	?	?	?	?	?	?	?	?	0.711	0.227

Therefore we got accuracy of nearly 71.1%, as the data input given is an image in MRI, the model can identify who has tumors and who doesn't, and this is very helpful for society.