1. **PROJECT REPORT ON MONUMENT CLASSIFICATION USING**

**CONVOLUTIONAL NEURAL NETWORK**

1. **PROJECT REPORT ON BREAST CANCER CLASSIFICATION USING SVM(SUPPORT VECTOR MACHINE) AND KNN(K-NEAREST-NEIGHBOUR)**

SUBMITTED TOWARDS PARTIAL FULFILLMENT OF PLACEMENT PREPARATION PROGRAM(2018-2019)

**SUBMITTED BY:**

SHREYA AGGARWAL (2010683)

GAGANPREET KAUR (2010635)

# ABSTRACT

We propose a model approach to classify Indian monuments according to their distinct architectural styles. While the historical significance of most Indian monuments is well documented, the details of their architectural styles are not as well recorded. Different Indian architectural styles often show certain similar features which makes classification a difficult task.

The model is trained on representations of different Indian monuments, obtained from cropped images, which exhibit geographic and cultural diversity. Experiments have been carried out on the manually acquired dataset that is composed of images of different monuments where each monument has images from different angular views. Therefore, we have curated a dataset of Indian monuments. In this paper, we propose two approaches to classify monuments according to their styles: Convolutional Neural Networks . The first approach gives an accuracy of 97%.

.

**INTRODUCTION**

A monument exists in the form of an object and also as symbol thereof. As a language symbol, a monument usually refers to something concrete, in some rare cases it is also used metaphorically. A monument can be a language symbol for a unity of several monuments or only for a single one, but in a broader sense it can also be used in nearly all knowable planes of being. What is considered a monument always depends on the importance it attributes to the prevailing or traditional consciousness of a specific historical and social situation.

With their elaborate superfluities and wonderful architecture, Indian monuments represent one of the most outstanding facets of the multi-faceted Indian culture. An architectural feat in itself, each Indian monument is a remarkably splendid sample of unbelievable artistry, covering a sense of mystery, deception and romance. Monuments are witnesses of India's past; the monuments of India are also the guardian pillars of India's cultural heritage. The monuments of India have become an inspiration for the future generations.

Monuments are also the tourist destinations in any country. They even are representations of great achievements present in art and architecture. It is therefore important to preserve them for the purpose that we can continue to enjoy their majestic views and the future generations too can learn from them. They are a part of India's vast heritage because they show the historical influence of any country with respect to its citizens.

Classifying a monument is difficult because many images of a single monument are to be used to train the system which are very much different from each other in their orientations. The differences in the images of monuments has been shown in Figure below which aptly shows the variations.



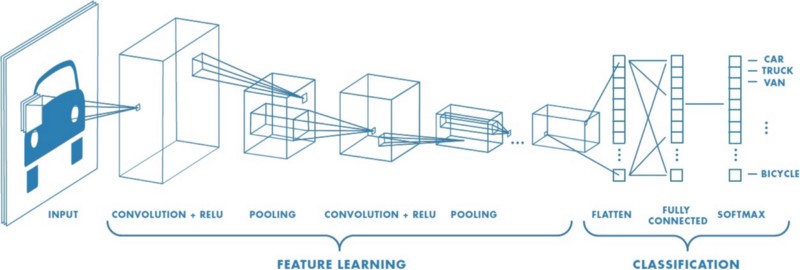
Monument recognition is a good concept though still not much work has been done in this domain of image classification. Moreover, the noise present in the images in the form of trees, people, animals, decorations etc. often leads to less accuracy. These variations make monument recognition a challenging problem.

**PROPOSED METHOD**

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.

The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.



Convolutional networks perceive images as volumes; i.e. three-dimensional objects, rather than flat canvases to be measured only by width and height. That’s because digital color images have a red-blue-green (RGB) encoding, mixing those three colors to produce the color spectrum humans perceive. A convolutional network ingests such images as three separate strata of color stacked one on top of the other.

So a convolutional network receives a normal color image as a rectangular box whose width and height are measured by the number of pixels along those dimensions, and whose depth is three layers deep, one for each letter in RGB. Those depth layers are referred to as channels.

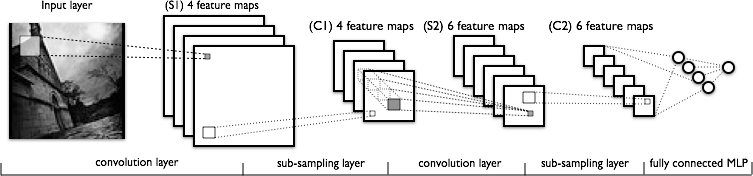
## Applications of CNN

CNNs are now used in almost every task in computer vision domain since it outperforms the older techniques if there is a significant amount of data available. Some applications are listed below :

* Object Detection ( Powers the Self driving cars)
* Face recognition
* Neural Art transfer
* X-ray diagnosis
* Satellite image analysis
* In astronomy

**Convolution Layer**

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.



**SOBEL FILTER**

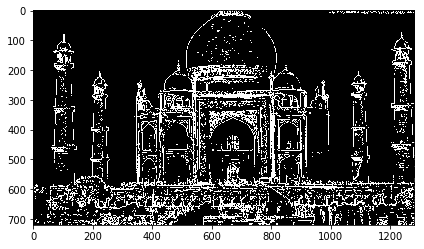
Sobel operator is used for edge detection. It is also used to detect two kinds of edges in an image:

* Vertical direction
* Horizontal direction

When we apply this mask on the image it prominent vertical edges. It simply works like as first order derivate and calculates the difference of pixel intensities in a edge region.

As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right and left pixel values around that edge. Also the center values of both the first and third column is 2 and -2 respectively.

This give more weight age to the pixel values around the edge region. This increase the edge intensity and it become enhanced comparatively to the original image.

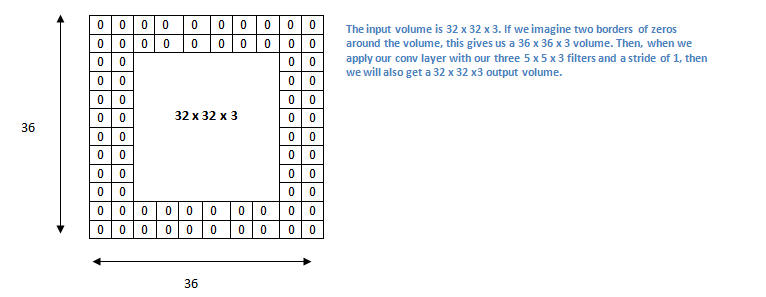
ORIGINAL IMAGE IMAGE AFTER SOBEL FILTER

#### Stride

Stride decides by how much we move our window ,when we have a stride of one we move across and down a single pixel. With higher stride values, we move large number of pixels at a time and hence produce smaller output volumes.

#### Padding

Padding is used to preserve the boundary information , since without padding they are only traversed once.



### Flattening Layer

This layer will convert the 3-dimensions (height,width,depth) into a single long vector to feed it to the fully connected layer or Dense layer. It connects every neuron in one layer to every neuron in another layer.

**EXPERIMENTAL SETUP:**

The section includes description of data used for testing and training. Moreover, the different experimental scenarios and their outcomes that were used for testing the performance of the model are also outlined in this section.

**DESCRIPTION OF DATASET:**

Our dataset comprises of 10 folder with each folder having 100 images per monument. The naming of each folder is done according to the name which corresponds to the monument. Famous Indian monuments like Taj Mahal, Qutub Minar, India Gate, etc are taken as dataset.

Our goal is to train a Convolutional Neural Network using Keras and deep learning to ***recognize*** and ***classify*** each of these monuments.

**WORKING OF OUR PROJECT:**

There are 3 directories:

1. dataset : Contains the five classes, each class is its own respective subdirectory to make parsing class labels easy.
2. examples : Contains images we’ll be using to test our CNN.
3. The training module: Contains our CNNModel  class .

There are 6 files in the root:

1. plot.png : Our training/testing accuracy and loss plot which is generated after the training script is ran.
2. lb.pickle : Our LabelBinarizer  serialized object file — this contains a class index to class name lookup mechanism.
3. md.model : This is our serialized Keras Convolutional Neural Network model file (i.e., the “weights file”).
4. train.py : We will use this script to train our Keras CNN, plot the accuracy/loss, and then serialize the CNN and label binarizer to disk.
5. classify.py : Our testing script
6. get-accuracy.py : Calculates overall accuracy.

**VGGNet-like architectures are characterized by:**

1. Using only 3×3 convolutional layers stacked on top of each other in increasing depth
2. Reducing volume size by max pooling
3. Fully-connected layers at the end of the network prior to a softmax classifier



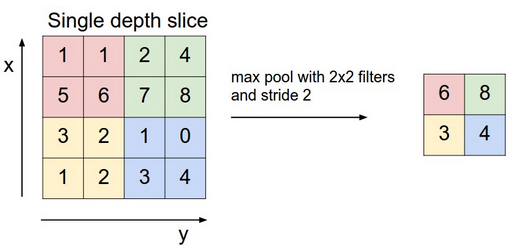
Rather than focus on one pixel at a time, a convolutional net takes in square patches of pixels and passes them through a filter. That filter is also a square matrix smaller than the image itself, and equal in size to the patch. It is also called a kernel, and its job is to find patterns in the pixels

A filter (or kernel) is an integral component of the layered architecture.

Generally, it refers to an operator applied to the entirety of the image such that it transforms the information encoded in the pixels. In practice, however, a kernel is a smaller-sized matrix in comparison to the input dimensions of the image, that consists of real valued entries.

## Max Pooling/Downsampling with CNNs:

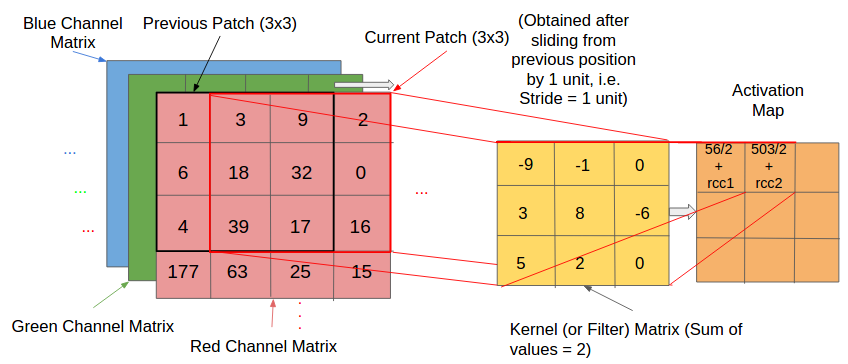
The next layer in a convolutional network has three names: max pooling, downsampling and subsampling. The activation maps are fed into a downsampling layer, and like convolutions, this method is applied one patch at a time. In this case, max pooling simply takes the largest value from one patch of an image, places it in a new matrix next to the max values from other patches, and discards the rest of the information contained in the activation maps.

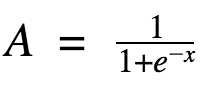


**ACTIVATION FUNCTION:**

It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).

The Activation Functions can be basically divided into 2 types-

1. Linear Activation Function
2. Non-linear Activation Functions
3. **Sigmoid or Logistic Activation Function(A) :**



The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict theprobability as an output. Since probability of anything exists only between the range of 0 and 1**,** sigmoid is the right choice.

The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points.

The **softmax function** is a more generalized logistic activation function which is used for multiclass classification.

1. **ReLU (Rectified Linear Unit) Activation Function:**

The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning.

**Range:**[ 0 to infinity)

The function and its derivative **both are** **monotonic**.

But the issue is that all the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly. That means any negative input given to the Relu activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.

**CODE EXPLANATION :**

🡪 The convolution layer has 32  filters with a 3 x 3  kernel. We’re using RELU  the activation function followed by batch normalization.

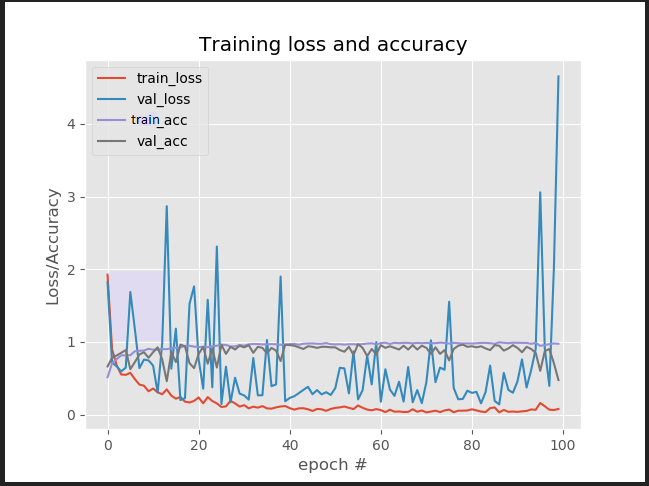
Our POOL  layer uses a 3 x 3  POOL  size to reduce spatial dimensions  96 x96  to 32 x 32 .

We’ll also be utilizing dropout in our network architecture. Dropout works by randomly disconnecting nodes from the *current layer* to the *next layer*. This process of random disconnects during training batches helps naturally introduce redundancy into the model — no one single node in the layer is responsible for predicting a certain class, object, edge, or corner.

🡪 The ImageDataGenerator  class will be used for data augmentation, a technique used to take existing images in our dataset and apply random transformations (rotations, shearing, etc.) to generate additional training data. Data augmentation helps prevent overfitting.

🡪 For our training script, we need to supply three required [command line arguments](https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/):

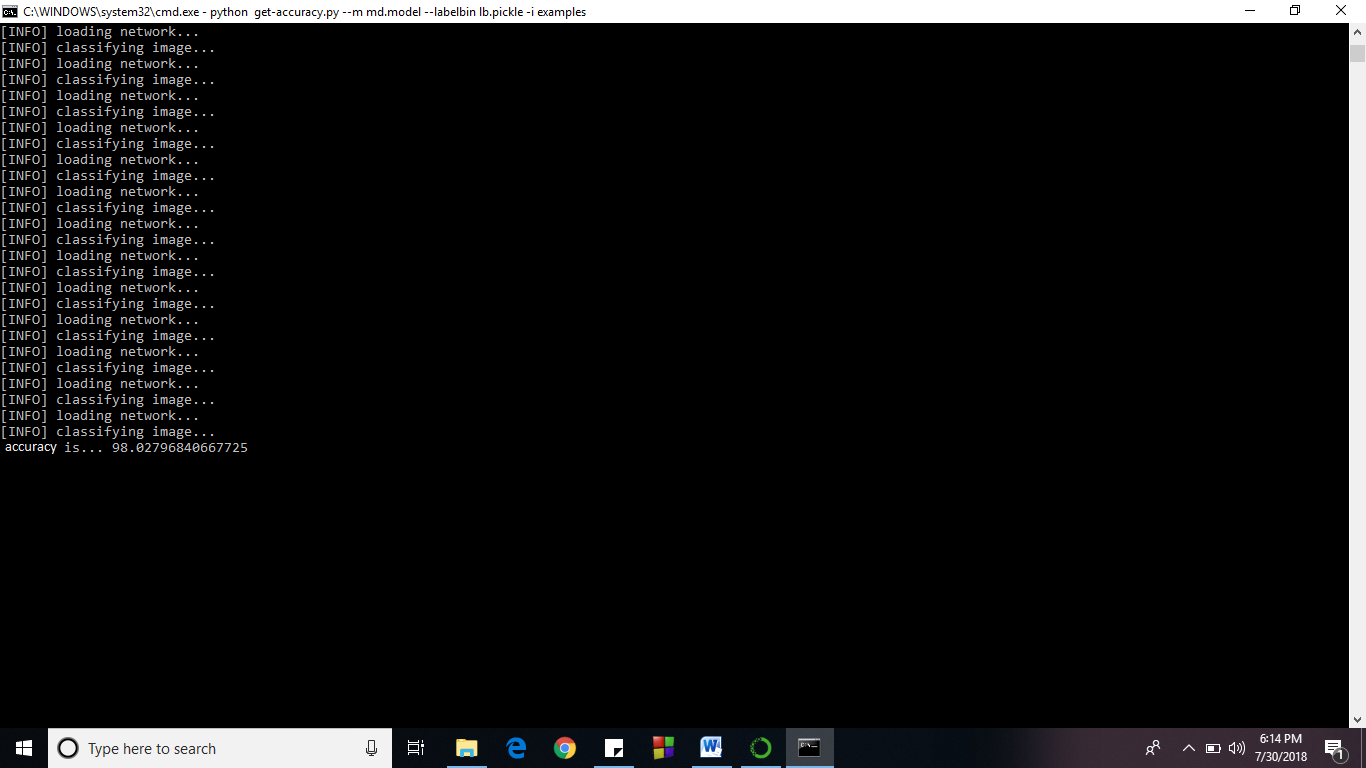
* --dataset : The path to the input dataset. Our dataset is organized in a dataset  directory with subdirectories representing each class. Inside each subdirectory is monument images.
* --model : The path to the output model — this training script will train the model and output it to disk.
* --labelbin : The path to the output label binarizer —we’ll extract the class labels from the dataset directory names and build the label binarizer.
* We also have one optional argument, --plot . If you don’t specify a path/filename, then aplot.png  file will be placed in the current working directory.



**THE TRAINING LOSS / ACCURACY PLOT**

This is the training loss-accuracy graph obtained after training the machine on the basis of dataset. We trained our machine with epochs = 100

**CONCLUSION:**

****

**REFERENCES:**

* <https://xrds.acm.org/blog/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/>
* <https://keras.io/models/sequential/>
* <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
* <https://en.wikipedia.org/wiki/Batch_normalization>
* <https://skymind.ai/wiki/multilayer-perceptron>
* <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

<http://www.everythingai.co.in/2018/06/09/convolutional-neural-networks-cnn/>

**2) BREAST CANCER CLASSIFICATION USING ORANGE TOOL**

**ABSTRACT**

In this project, we have done breast cancer classification using KNN. Cancer diagnosis is one of the most studied problems in the medical domain. Breast cancer is one of cancer killer in the world. The diagnosis of this cancer is a big problem in cancer diagnosis researches. In artificial intelligent, machine learning is a discipline which allows to the machine to evolve through a process. One of the most popular methods is K-nearest neighbors (K-NN) which is a supervised learning method. The quality of the results depends largely on the distance and the value of the parameter “k” which represent the number of the nearest neighbors.

Support vector machine (SVM) and K-Nearest Neighbor (KNN) classifier is a combined classifying method, which has excellent performance for various applications. The purpose of this project is to examine the performance of the SVM-KNN classifier on the diagnosis of breast cancer using tumor dataset. The objective is to classify the recurrence and non-recurrence of breast cancer. Experimental results show that SVM-KNN model has achieved a performance with 80.06% classification accuracy on testing subset.

**INTRODUCTION OF TOOL USED**

 Orange is a Python library that powers Python scripts with its rich compilation of mining and machine learning algorithms for data pre-processing, classification, modelling, regression, clustering and other miscellaneous functions. Orange also comes with a visual programming environment and its workbench consists of tools for importing data, and dragging and dropping widgets and links to connect different widgets for completing the workflow.

Orange is often a component structured data mining as well as machine learning software suite created in the python language. It's a good open source data visualization as well as evaluation with regard to novice and experts. Data mining can be done via visual programming or even python scripting. It's components with regard to machine learning.

Orange is often a quite capable open source visualisation as well as group of data mining tools along with a user friendly. Many analyses is possible via its visual programming interface (drag and drop associated with widgets) and many visual tools tend to be supported such as scatterplots, bar charts, trees, dendograms and heatmaps. A significant number (more than 100) associated with widgets tend to be supported.

As the name identify, it's a good open source tool aiming to make data mining “fruitful and fun”. It provides data mining with the use of visuals. It can be data visualizations along with analysis created possible for novice users in addition to with regard to experts. Users can easily layout info analyses by means of visual programming and also Python scripting.

Orange believes data mining needs to be ‘fruitful and fun’, whether you have years of experience or are just getting started in the discipline. There is visual programming along with Python scripting tools pertaining to data visualizations and analysis. Its operating system is windows, linux, OS X.

In orange, data analysis procedure could be created via visual programming. Orange remember the choice, propose most frequently used combination. Orange offers functions with regard to various visualizations, for example scatterplots, bar charts, tree, to dendrograms, network as well as heatmaps.

## Orange Characteristics:

* Wrappers with regard to scikit-learn algorithms.
* Numpy dependent data storage.
* Improved data pre-processing.
* Focus on interactive visualization.
* SQL data can easily stick to the particular database server.
* Orange reads Google Sheets.
* Color Your Data.

**SUPPORT VECTOR MACHINE:**

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables a dummy variable is created with case values as either 0 or 1. Thus, a categorical dependent variable consisting of three levels, say (A, B, C), is represented by a set of three dummy variables:

A: {1 0 0}, B: {0 1 0}, C: {0 0 1}

To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function. According to the form of the error function, SVM models can be classified into four distinct groups:

* Classification SVM Type 1 (also known as C-SVM classification)
* Classification SVM Type 2 (also known as nu-SVM classification)
* Regression SVM Type 1 (also known as epsilon-SVM regression)
* Regression SVM Type 2 (also known as nu-SVM regression)

**KNN:**

**KNN** is a **non-parametric, lazy**learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

KNN can be used for **classification** — the output is a class membership (predicts a class — a discrete value). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It can also be used for **regression** — output is the value for the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbors.

### ****Some pros and cons of KNN****

**Pros**:

* No assumptions about data — useful, for example, for nonlinear data
* Simple algorithm — to explain and understand/interpret
* High accuracy (relatively) — it is pretty high but not competitive in comparison to better supervised learning models
* Versatile — useful for classification or regression

**Cons**:

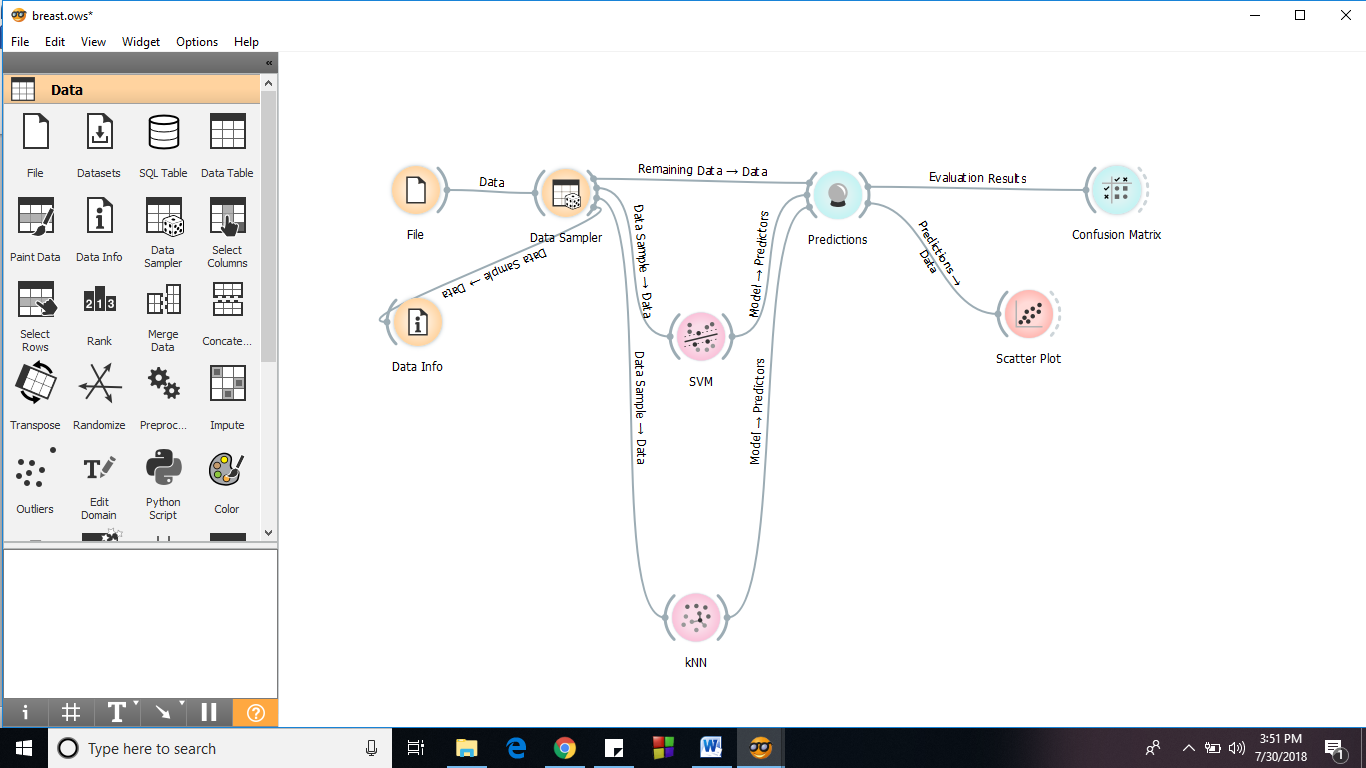
* Computationally expensive — because the algorithm stores all of the training data
* High memory requirement
* Stores all (or almost all) of the training data
* Prediction stage might be slow (with big N)
* Sensitive to irrelevant features and the scale of the data

**ABOUT OUR DATASET:**

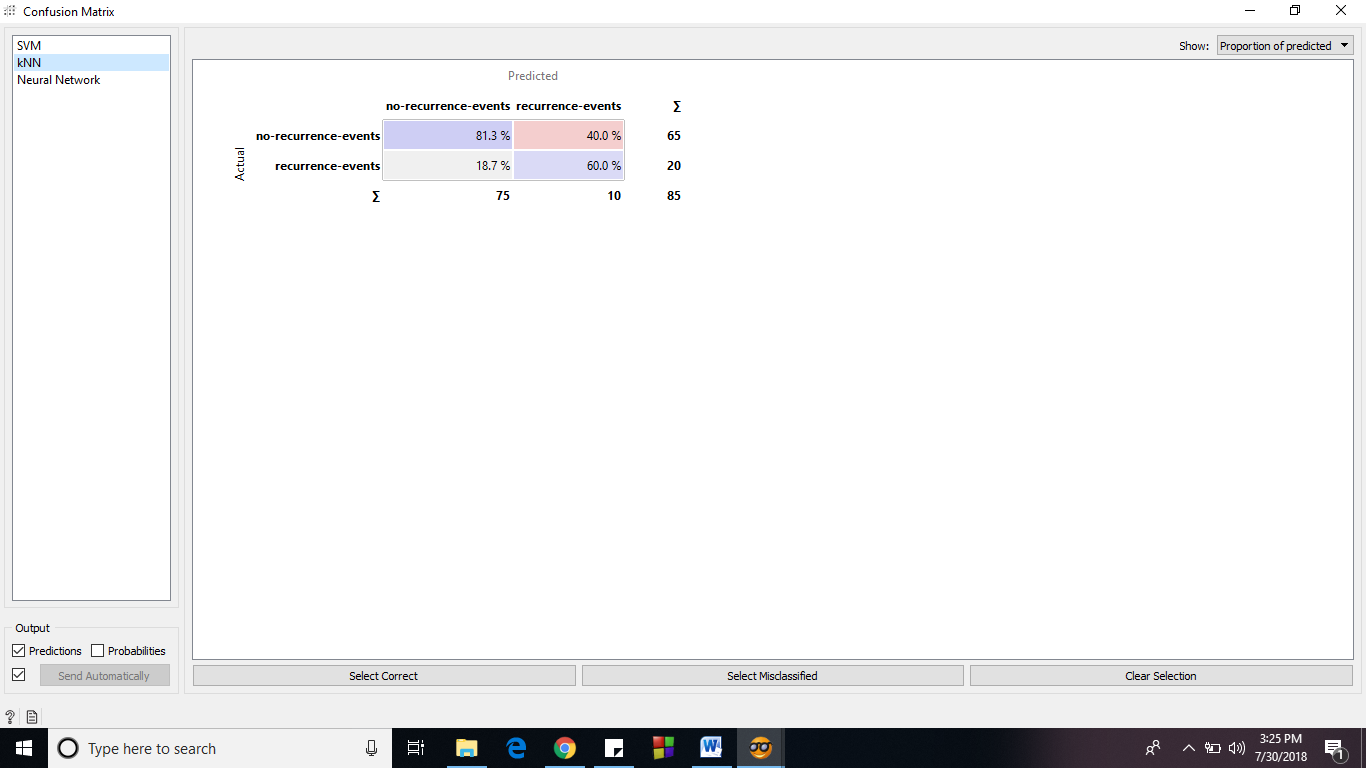
This data set includes 201 instances of one class and 85 instances of another class. The instances are described by 7 attributes, some of which are linear and some are nominal.

**Attribute Information:**

Class: no-recurrence-events, recurrence-events   
1. age  
2. menopause   
3. tumor-size  
4. inv-nodes  
5. node-caps  
6. deg-malig  
7. breast

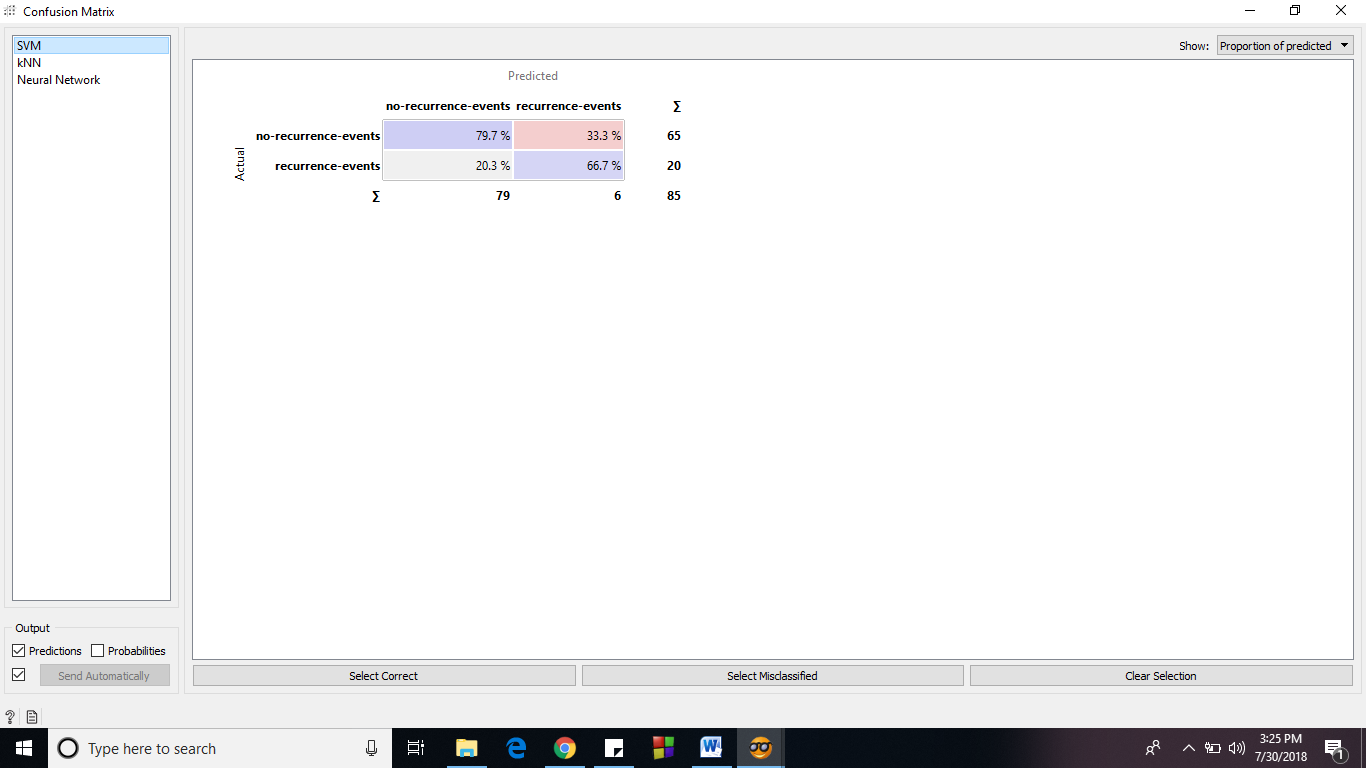
****

* **Data**: widgets for data input, data filtering, sampling, imputation, feature manipulation and feature selection
* **Visualize**: widgets for common visualization .We have used Scatter plot.
* **Classify**: a set of supervised machine learning algorithms for classification
* **Regression**: a set of supervised machine learning algorithms for regression
* **Evaluate**: cross-validation, sampling-based procedures, reliability estimation and scoring of prediction methods. For evaluation, we have used Confusion matrix and prediction widgets.

**RESULTS**

USING KNN

USING SVM



SCATTER PLOT

