Comparative Analysis of Different ML & DL Methods on Brain Image Classification

Jared Woolsey
jawoolsey@ucdavis.edu
UC Davis

Pranav Ramesh
prramesh@ucdavis.edu
UC Davis

Elsie Basa
efbasa@ucdavis.edu
UC Davis

April Vang
apvang@ucdavis.edu
UC Davis

Sehee Han
sehhan@ucdavis.edu
UC Davis

1 Introduction

Brain tumors, which can be either cancerous (malignant) or non-cancerous (benign), require accurate detection and classification for early diagnosis and effective treatment planning. In this regard, MRI machines play a crucial role by capturing detailed images of the brain, enabling healthcare professionals to identify and characterize brain tumors, thereby facilitating informed decision-making and personalized care. In order to enhance the performance of MRI machines, it is imperative to employ the most efficient learning algorithm within the system. The main purpose of this analysis is to examine and compare the performance of learning algorithms such as Convolution Neural Networks (CNN), Support Vector Machines (SVM), Logistic Regression, and Random Forest by predicting the malignancy of brain tumors.

Convolution Neural Networks (CNN) are a class of deep learning algorithms specifically designed for analyzing visual data, such as images or videos. They have gained significant attention in medical imaging, due to their ability to automatically learn and extract meaningful features from raw input data. Support Vector Machines (SVM) is a machine learning algorithm for classification and it works by finding an optimal separation between different classes by selecting a hyperplane with the maximum margin in a high-dimensional feature space. SMV offers certain advantages over CNNs in terms of handling high-dimensional feature spaces, generalizing efficiently with limited training data, and providing clear decision boundaries. Additionally, random forests will be applied. In image classification using a random forest, multiple decision trees are trained on various image features, and the final classification is determined based on the collective votes of the trees, leveraging their ability to discern patterns and distinguish between different classes.

This analysis dives into the effectiveness of the proposed models for predicting malignant tumors. Mentioned algorithms above will be used and their performance will be evaluated and compared against each other. The performance of each algorithm will be assessed based on key metrics such as accuracy, precision, recall, and F1 score. By comparing SMV, logistic regression, random forest, and CNN, we aim to gain insight into their respective capabilities in image classification. This analysis aims to contribute to the development of robust and reliable diagnostic systems for neuroscience and oncology applications.

2 Problem Definition, Data Description and Literature Review

2.1 Probelm Definition

The purpose of this analysis is to examine and compare various algorithms in terms of their ability to predict an MRI image dataset using multiple metrics. Additionally, we will compare whether machine learning and deep learning algorithms show a remarkable difference in performance.

2.2 Data Description

The dataset consists of a total of 253 grayscale MRI images, out of which 155 images depict brain tumors, while the remaining 98 images represent brain samples without any tumor manifestation. Regrettably, the Kaggle source does not provide any accompanying information regarding the origin or source of the MRI scans. Despite the absence of specific details, the availability of image data still enables the implementation of the model for brain tumor prediction. The MRI images utilized in the training of the model depict late-stage malignant brain tumors, with predefined categories labeled as "Yes" to indicate the presence of a brain tumor and "No" to denote its absence.

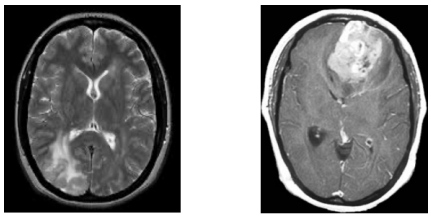


Figure 1: Left: Image with no Brain Tumor, Right: Image with Brain Tumor

For the data preparation for the machine learning models, images are flattened into feature vectors; feature extraction to enhance classification performance; feature scaling to normalize the data; training the model to learn the relationship between features and labels; and finally, using the trained model to predict class labels for new images by computing the criterion for selecting the class with the highest probability.

2.3 Literature Review

Here, we review past efforts of classification on brain tumor MRI datasets. We will start with analyses utilizing our deep learning model and then proceed with the machine learning models.

2.3.1 Deep Learning - CNN

Using the GoogleNet and AlexNet architecture, an accuracy of 99.45% and 98.95%, respectively, as well as a sensitivity of 99.75% and 98.4% were found. AlexNet uses 8 layers and takes 60 million parameters (Swarup et al. 2023). Three group classification has been performed identifying images as either cropped, uncropped, or segmented lesions, still with an accuracy of 98.93%, 99%, and 97.62% respectively, and a sensitivity of 98.18%,98.52%, and 97.40% respectively (Alqudah 2019). Using the VGG16 and "23-layer CNN" architectures, models achieved 97.8% and 100% classification accuracy respectively (100 percent was achieved on a smaller dataset with only 121 images). This was using multi-class classification, and a 70/30 train-test split (Khan et al. 2022). Utilizing a five-layer CNN algorithm with a 70/30 to 80/20 train-test split but for binary classification, 97.8% accuracy was attained, much better than the corresponding 70/30 split which had a 92.98% accuracy (Hossain et al. 2019).

2.3.2 Machine Learning- SVM, Logistic Regression & Random Forest

Using a random forest for binary classification, 89.39% accuracy, 98.3% recall, and 90.3% precision were recorded. In the same study, SVM attained 92.42% accuracy, 98.3% recall, and 93.5% precision.

On the other hand, in another study (Ramdas Vankdothu 2022), SVM showed the worst performance and a wide variation in accuracy rates when cross-validation was performed. The best accuracy rate was 0.81, while the worst accuracy rate was 30%. The mean accuracy rate was approximately 50%, indicating that the model did not exhibit consistently high predictive performance. The mean sensitivity was 65%, and the mean specificity was 39%. Logistic Regression had 87.88% accuracy, 94.9% recall, and 91.8% precision (Hossain et al. 2019). Through a different type of logistic regression and threshold segmentation on the ADNI-1 & ADNI-2 datasets, 97% accuracy, 97% recall, and 97.9% precision were attained. (Gajula and Rajesh 2022)

3 Proposed Method

3.1 Intuition

CNN are shown to be effective for image detection due to their ability to learn and extract features from images. Based on the fact that CNN is a deep learning algorithm that is specifically designed for image classification, we predict that CNN will perform the best of all the artificial intelligence algorithms. Next, our motivation for applying an SVM model is that it has also been shown to perform suitably well for binary image classification (here we used training size 202x50176) in a high-dimensional setting to distinguish between tumor or no tumor. SVM flexibly accepts different kernel functions if we believe a linear hyperplane will not classify well. Support vectors additionally enable a robust model to address overfitting as motivation for applying this method to our dataset. Logistic regression can be applied to brain tumor detection by incorporating relevant features extracted from the images. The idea is that the logistic regression model may exhibit discriminative patterns that can distinguish between tumor and non-tumor samples. While logistic regression may not inherently capture complex visual patterns like CNNs, its simplicity and interpretability can still provide insights into chosen features. Random Forests are robust to noise and outliers since they consider multiple randomly sampled subsets of features and training samples when constructing individual trees. This inherent randomness in the model helps reduce the impact of noisy data, making Random Forests suitable for handling the inherent variability in brain tumor images. Also, this algorithm can provide advanced interpretability through feature importance estimation.

3.2 Description of its Algorithms

3.2.1 CNN

A CNN consists of multiple interconnected layers that perform specific operations on input data. The key components of a typical CNN architecture include convolutional layers, pooling layers, fully connected layers, and an output layer. Convolutional Layers: These layers apply a set of learnable filters or kernels to the input data, convolving them across the entire input volume to extract spatially localized features. The convolution operation computes dot products between the filter weights and the input, resulting in feature maps that highlight relevant patterns in the data. Pooling Layers: Pooling layers reduce the spatial dimensions of the feature maps while retaining essential information. Max pooling and average pooling are commonly used operations in CNNs, which downsample the feature maps by selecting the maximum or average value within a predefined window. Fully Connected Layers: Fully connected layers connect every neuron from the previous layer to every neuron in the subsequent layer, enabling high-level feature extraction and classification. These layers perform non-linear transformations on the input data to capture complex relationships between features. Output Layer: The output layer is responsible for producing the final predictions or classifications based on the features extracted by the previous layers. The activation function used in the output layer depends on the task at hand, such as softmax for multi-class classification or sigmoid for binary classification. (Indolia et al. 2018)

The algorithm for training a CNN involves the following steps:

Firstly, during the data preprocessing process, input data, such as brain tumor images, are typically preprocessed to enhance their quality and facilitate network training. Common preprocessing steps include resizing, normalization, and augmentation techniques to increase the diversity of the training data. Nextly, during forward propagation, the input data is passed through the network layer by layer. Each layer applies its specific operations, such as convolutions and pooling, to transform the input data and generate increasingly abstract representations of the input features. After the forward propagation, the network's output is compared to the ground truth labels to calculate the loss. The choice of loss function depends on the task, such as cross-entropy loss for classification problems.Then, backpropagation is used to update the weights of the network to minimize the loss. The gradients of the loss with respect to the network parameters are computed and propagated backward through the layers. This process allows the network to learn from its mistakes and adjust the weights accordingly. Nextly, optimization algorithms, such as stochastic gradient descent (SGD) or its variants like Adam or RMSprop, are used to update the network weights based on the computed gradients. These algorithms iteratively adjust the weights to find the optimal set of parameters that minimize the loss. Lastly, training and validation steps are conducted. The training process involves repeatedly iterating over the training dataset, performing forward and backward propagation, and updating the weights. The model's performance is evaluated using a separate validation dataset, and training can be stopped based on predefined stopping criteria, such as convergence or early stopping techniques.

Transfer learning is a technique commonly used in CNNs, particularly when the available labeled data is limited. Pre-trained CNN models, trained on large-scale datasets such as ImageNet, are utilized as a starting point. The pre-trained model's weights and learned features are then fine-tuned using a smaller dataset specific to the brain tumor detection or classification task. Fine-tuning helps leverage the knowledge gained from a larger dataset and accelerate the learning process for the specific task at hand. We choose softmax since it gives the predicted probability that the image belongs to a particular class.

3.2.2 SVM

SVM is based on the principle of structural risk minimization rather than an empirical risk minimization. Input data are viewed as a p dimensional vector and SVM tries to separate the points using a $p-1$ dimensional hyperplane. There are many hyperplanes which can be used for the classification process. To determine the best hyperplane, SVM searches through the high-dimensional space for sets of hyperplanes to find the one that represents the largest margin between the two tumor classifications, and we want the distance from the nearest data point on each side to be maximized. Support vectors are used to define the hyperplane for generating predictions. To obtain the maximum-margin hyperplane, we consider looking at different kernel functions as oftentimes datasets are not linearly separable in that space and so we construct nonlinear classifiers to apply the kernel trick, $K(x_i, x_j)$. In the example, we examine the following three kernels (1) linear kernel $K(x_i, x_j) =< x_i, x_j >$, (2) polynomial kernel $K(x_i, x_j) = (c_0 + \gamma < x_i, x_j >)^d$, and (3) radial basis function (rbf) kernel: $K(x_i, x_j) = exp(-\gamma||x_i, x_j||^2)$. Next, we performed GridSearchCV to obtain the optimal hyperparameters for our SVM models. The grid is as follows: C: [0.05, 0.1, 0.15, 0.25, 0.5, 0.75, 1], gamma: [0.5, 1], kernel: [linear, poly, rbf], degree: [2, 3, 4], random state: [0]. After searching through exhaustively, he optimal parameters are: 'C': 0.05, 'degree': 4, 'gamma': 1, 'kernel': 'poly', 'random state': 0.

3.2.3 Logistic Regression

The logistic function is a sigmoid function and outputs values between [0,1]. We use $p(x) = \sigma(t) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$ where μ is the location parameter midpoint, s is a scale parameter,$\beta_0 = -\mu/s$ is the intercept of the line $y = \beta_0 + \beta_1 x$. Next, we define the logit, or log odds, function as the inverse σ^{-1} of the standard logistic function with a simple form as follows:

$$g(p(x)) = \sigma^{-1}(p(x)) = logit(p(x) = \frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 x}$$

The objective is to minimize the negative log-likelihood or cross-entropy loss and we use the coordinate descent to iteratively update the weights to minimize the loss function and set a 'liblinear' solver. This solver is well-suited for small to medium-sized datasets and works efficiently for binary classification problems. It employs a linear model with a regularization term to prevent overfitting and achieve better generalization. Next, GridSearchCV is used with 'C' values, C': [0.1, 1.0, 5], evaluating each combination using cross-validation. The hyperparmeter 'C' that was selected was C= 0.1.

3.2.4 Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees through feature and sample subsets, utilizing majority voting for predictions. It handles high-dimensional data, mitigates overfitting, and provides accurate results. Random Forest is widely used in classification and regression tasks, offering valuable insights into feature importance. In image classification, Random Forest is a powerful algorithm that leverages the collective knowledge of multiple decision trees to accurately classify images, handling high-dimensional image data and providing valuable insights into feature importance. Random Forest uses a bootstrapping procedure to obtain a random subset of features during the learning process which is what ultimately leads to better model performance as it lowers the variance without increasing bias.

A grid search was performed for Random Forest for classification, exploring different hyperparameter values to find the optimal combination that yields the best performance. The hyper-parameters tested were 'n estimators' (with a value of 200), 'criterion' (with options 'gini' and 'entropy'), min samples split' (with values 2 and 5), and 'min samples leaf' (with a value of 1). The 'gini' criterion minimizes misclassification by measuring node impurity, while the 'entropy' criterion maximizes information gain by reducing uncertainty in node splits, although it can be more sensitive to outliers and lead to balanced splits. The optimal one chosen was criterion ='entropy', min samples leaf = 1, min samples split = 2, and n estimators = 200.

4 Data Analysis

4.1 Model Assumption

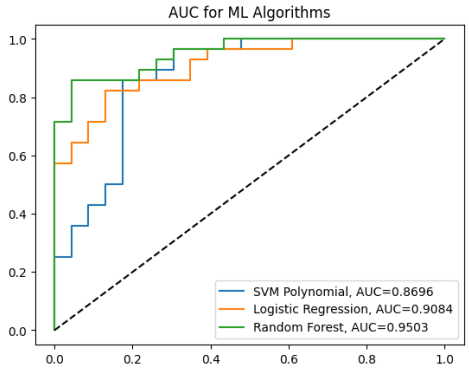
Each machine learning algorithm operates based on its specific assumptions. For CNN, assume the presence of local spatial correlations, translation invariance, and hierarchical representations in the input data. SVM assumes linear separability and feature independence, utilizing the kernel trick to enhance separability. Logistic Regression assumes a linear relationship between input features and log odds, independence of errors, and absence of multicollinearity. Random Forest assumes feature independence within decision trees, incorporates randomness in feature selection, and relies on majority voting. These assumptions form the foundation of the algorithms, shaping their behavior and influencing the predictions they make. One major advantage of using CNN was its extremely relaxed assumption requirements. There is no major risk of violating the assumptions, thus making it a consistently reliable approach to employ CNN for the given task.

4.2 Questions of Interest and Figures

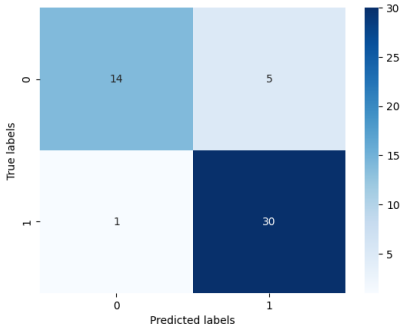
We have a few primary research questions of interest:

1. Perform classification using Machine Learning and Deep Learning algorithms, including both machine learning and deep learning algorithms. We want to achieve as high prediction accuracy on the testing set as possible

2. Compare the performances of CNN, SVM, Random Forest, and Logistic Regression algorithms in terms of testing accuracy, precision, recall, F-1 score, etc.
3. Determine which algorithm of the previously mentioned is the best overall performance.



(a) Comparison of AUC for each ML Algorithm



(b) Confusion Matrix of CNN

Figure 2: AUC for ML algorithms and CNN Confusion Matrix

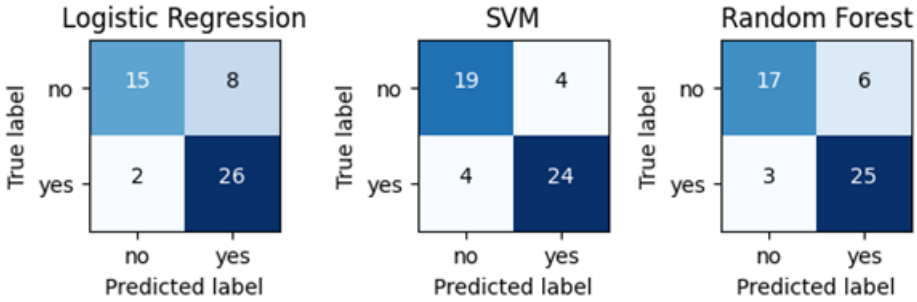


Figure 3: Confusion Matrices of SVM, Random Forest, and Logistic Regression

	SVM	Log Reg	Rand Forest	CNN
Precision	0.857	0.765	0.806	0.857
Recall	0.857	0.929	0.893	0.967
Accuracy	0.843	0.804	0.824	0.88
F1 Score	0.857	0.839	0.847	0.909

Table 1: Accuracy, Precision, Recall, and F1 Score of ML Algorithms

4.2.1 Machine Learning: SVM, Logistic Regression, Random Forest

SVM: The F-1 score for SVM was higher than that of any other ML algorithm, due to the overall high precision and recall combination. We also used a polynomial kernel instead of a linear kernel found through grid search which gives improvement in results.

Logistic Regression: Logistic regression performed quite well, considering expectations. Of the ML algorithms, Logistic regression had the highest recall (TPR), even higher than CNN at 90%. This is quite an important metric to score highly on, since if someone has a brain tumor, it will nearly always classify it correctly. However, with this comes the worst precision overall at 76.50%. All the other matrices had precision rates of over 80%. In terms of the ROC curve, measuring at 90.84% of the Area Under the Curve (AUC) was the third best overall. Accuracy and F1 score were also lower than all other models. Overall, even though Logistic regression performed the worst, it was quite simple to implement and served as a basis for a lower bound on what to expect for other algorithms.

Random Forest: The Random Forest model demonstrated exceptional performance across the overall parameters. It had the highest AUC among all the algorithms at 95.03%. However, this is countered by the fact that it had a poor precision of 80.6% and poor accuracy of 82.4%. We also include a decision tree for our random forest, which can be seen in Figure 4 in the Appendix section.

4.2.2 Deep Learning - CNN

Utilizing a 9-layer CNN model, with 100 epochs, we obtained an accuracy of 88%, precision of 85.7%, a recall of 96.7%, and a total F1 score of 90.9% on the testing set. These results were much better than the ML algorithms on all metrics and perform worse than all ML's AUC, which measured in at 85%. This shows that CNN is overall the best algorithm to run for image recognition/classification, which makes sense because CNN is designed primarily for image classification. The CNN model is given in Figure 5 in the Appendix section.

5 Conclusion

5.1 Limitations

SVM is rather sensitive to our grid search when selecting optimal hyperparameters depending on our test and training size which makes it challenging to determine how well our SVM model will generalize to additional external validation data. Logistic regression does not inherently consider spatial information and local structures in images, which can limit its ability to capture certain details and spatial relationships which are important for classifying image data. It could likely be why it has the lowest accuracy rate. There are very few limitations to CNN, as image classification is a primary goal of the algorithm, but it should be noted that sufficient data is necessary to achieve high accuracy rates. Although this is an advantage in considering model assumptions, CNN also faces the inherent issue of a lack of interpretability. This is to say that we do not know which features are considered, so it can be quite difficult to learn about how to detect brain tumors using non-CNN algorithms. Random Forest might have trouble capturing complicated and fine-grained details in complex image datasets due to the limited consideration of features by individual decision trees which could potentially result in the loss of subtle visual patterns. Also, there is a risk of overfitting with this model given the fact that the limitation of insufficient amounts of diverse data is present in our dataset (only 253 images), so it is important to exercise appropriate critical judgment in parsing the meaning of our conclusions. With a higher sample size, it would be possible to break apart our input into further sub-classifications, which may lead to enhanced results for certain algorithms.

5.2 Conclusion

Overall, most of the Machine Learning, Random Forest, SVM, and Logistic Regression performed similarly. Out of the three algorithms, SVM had the best precision and F1 score, both at 85.7%, and had the highest ML accuracy of 84.3%, which exhibited even better performance than some of our literature reviews that utilized SVM. Though we had better results for SVM, we highlight that it is not a robust model for our dataset and sensitive to parameter changes. Random Forest had the best recall at 89.3%. Logistic Regression performed the best in terms of true positive rate (recall = 0.929). In the context of identifying brain tumors, this is a very important metric, as it represents how often a brain tumor is spotted. Mistakenly identifying a brain tumor when there is not one is not as crucial of an issue due to the fact that the mistake would not result in death, dissimilar to how failing to spot a tumor may be catastrophic.

In the end, CNN outperformed all the Machine Learning models in all metrics, except the AUC. It had better accuracy, F1 score, precision, and recall rates. It did have a very low AUC of about 85%, which is lower than all the ML models. This paper demonstrated that overall deep learning algorithms have advantages over machine learning algorithms. However, it is to be noted that the run time for deep learning algorithms is significantly longer compared to the machine learning ones. The 9-layer CNN that was utilized in this paper took more than 2 hours to run compared to about 1-5

minutes for the machine learning methods. By reducing the number of layers the runtime could be shortened, however, the results might ultimately be less accurate. This is due to the fact that adding layers to CNN enhances its ability to discern patterns and certain features. Despite the increased computational time, the advantages of deep learning, such as improved accuracy and the ability to automatically learn intricate features, make them a valuable choice for the classification of images.

6 References

Alqudah, Ali Mohammad (Dec. 2019). "Brain Tumor Classification Using Deep Learning Technique - A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes". In: *International Journal of Advanced Trends in Computer Science and Engineering* 8.6, pp. 3684–3691. DOI: 10.30534/ijatcse/2019/155862019. URL: https://doi.org/10.30534%2Fijatcse%2F2019%2F155862019.

Gajula, Srinivasarao and V. Rajesh (2022). *An MRI brain tumour detection using logistic regression-based machine learning model*. DOI: https://doi.org/10.1007/s13198-022-01680-8.

Hossain, Tonmoy et al. (2019). "Brain Tumor Detection Using Convolutional Neural Network". In: *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–6. DOI: 10.1109/ICASERT.2019.8934561.

Indolia, Sakshi et al. (2018). "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach". In: *Procedia Computer Science* 132. International Conference on Computational Intelligence and Data Science, pp. 679–688. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.069. URL: https://www.sciencedirect.com/science/article/pii/S1877050918308019.

Khan, Md. Saikat Islam et al. (2022). "Accurate brain tumor detection using deep convolutional neural network". In: *Computational and Structural Biotechnology Journal* 20, pp. 4733–4745. ISSN: 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2022.08.039. URL: https://www.sciencedirect.com/science/article/pii/S2001037022003737.

Ramdas Vankdothu, Mohd Abdul Hameed (2022). "Brain tumor segmentation of MR images using SVM and fuzzy classifier in machine learning". In: 31, p. 9. URL: https://www.sciencedirect.com/science/article/pii/S2665917422000745?via%3Dihub.

Swarup, Chetan et al. (2023). "Brain tumor detection using CNN, AlexNet amp; GoogLeNet ensembling learning approaches". In: *Electronic Research Archive* 31.5, pp. 2900–2924. ISSN: 2688-1594. DOI: 10.3934/era.2023146. URL: https://www.aimspress.com/article/doi/10.3934/era.2023146.

7 Appendix

7.1 Dataset and Codes

Please see the dataset and codes within the zipped file.

7.2 Supplementary Figures

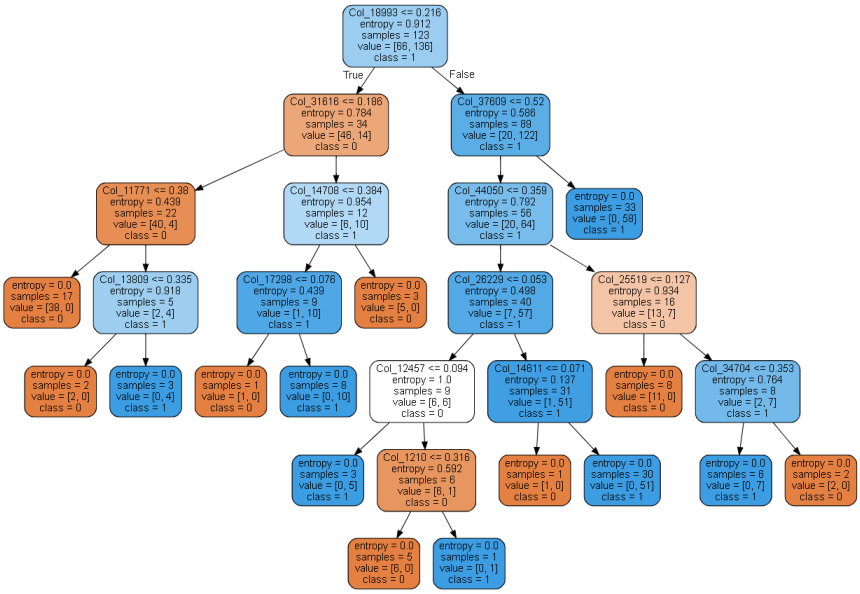


Figure 4: Decision Tree for the Random Forest algorithm

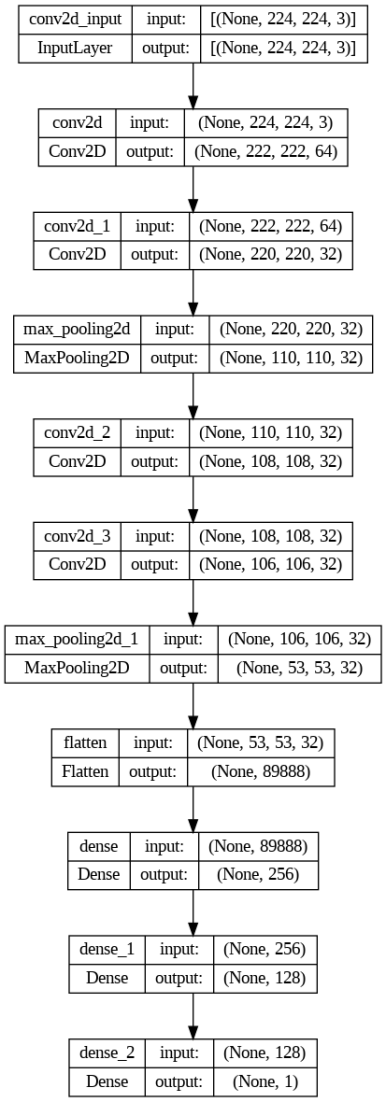


Figure 5: CNN Layers Design