**UNIVERSITY OF TEXAS AT ARLINGTON**

A logo of a university of texas

Description automatically generated

**CSE 6363: - Machine Learning**

***MELANOMA DETECTION BASED ON DEEP NEURAL NETWORKS***

**By**

**Project Team 4**

**Niranjana Subramanian(UTA ID: 1002046305)**

**Rohan Rajan Vaidya (UTA ID: 1001987656)**

**Sai Venkat Reddy Kopparthi(UTA ID:** **1002113962 )**

**Phase- 3**

**CONVOLUTIONAL NEURAL NETWORK**

Neural networks are the significant learning algorithms that come at the centre of deep learning algorithms. They are part of a wide array of machine-learning branches. Overall, they contain input layers, a few hidden layers, and an output layer known as node layers. Each node in the neural network has an associated weight and threshold, and each node is connected to another node. The activation of an individual node in the network depends on whether the output of that specific node is above the threshold. If such is the case, then that node begins transferring data to the next network tier. Otherwise, no data is sent to the next level of the network. The convolutional neural networks work with classification and on a computer vision task.

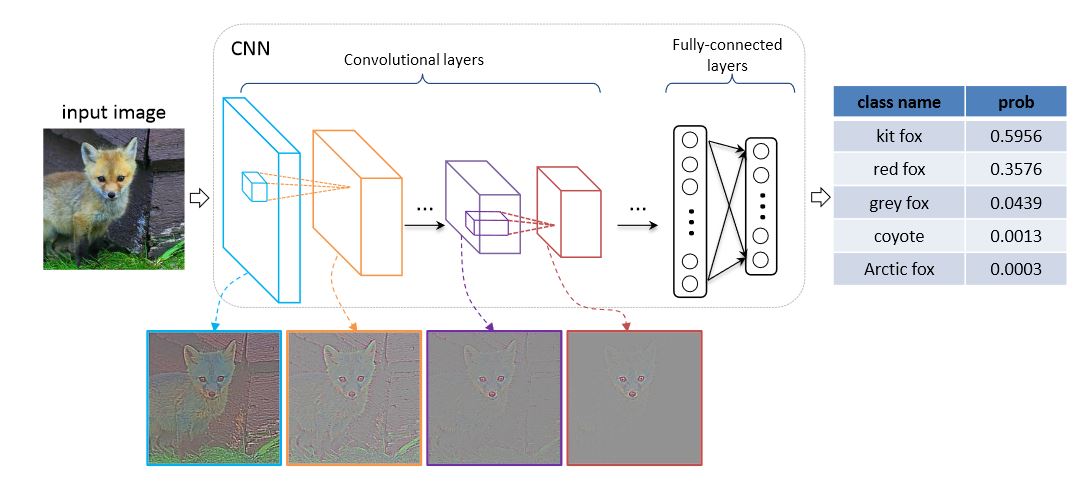
The CNN's have three main types of layers:

1. **Convolutional layer**: The convolution layer is the nucleus building block of CNN, where most of the computation occurs. It needs input data, a feature map, and a filter to perform the computations.
2. **Pooling Layers**: The pooling layers are down-sampling layers that perform dimensionality reduction. The difference between the convolution layer and the pooling layer is that the filter does not have any weights in the pooling layer, and the kernel applies the aggregation function to the values in the receptive field.
3. **Fully** **Connected Layer**: The fully connected layer deals with the task of classification based on the inputs received from previously connected layers.

**VISUAL GEOMETRY GROUP ARCHITECTURE**

The architecture we have used for building our deep convolutional neural network model is VGG. It refers to the layers we have used in our model, which are 16. Hence, we have used the VGG-16 model. VGG are convolutional neural networks (CNNs) which are powerful feature extractors. They capture hierarchical and abstract features from images through convolutional and pooling layers.​ After the feature extraction, the fully connected layers (dense layers) in the traditional head of the neural network learn to map these features to the final classes.

**The VGG Architectural Diagram:**



**MODEL OVERVIEW**

A screenshot of a computer code

Description automatically generated

We initialize the VGG 16 model with arguments:

1. **Weight**: This argument is used for preloading the weights from the ImageNet dataset which act as a firm starting point for extracting features.
2. **Include\_top**: We have set this argument to false as this allows us to add our own custom layers on the top, to work well with our melanoma dataset.
3. **Input\_shape**: The input shape argument allows input to be a three-channel image with 224x244 pixels.

**Custom Layers:**

1. **Flatten**: This layer flattens output received from the VGG 16 model. The output of the VGG model is a 3D tensor, and hence, we flattened it to establish a connection to the fully connected layer.
2. **Dense(1024, activation='relu'):** This is the fully connected layer that contains 1024 units and ReLU activation. We have utilized this hidden layer as it enables learning of complex features from the flattened data.
3. **Dense(512, activation='relu'):** This layer is similar to the above but has 512 units.
4. **Dense(2, activation='softmax'**): The final dense layer has two units because it is a binary classification task. The softmax activation function is used as the classifier, producing the class probabilities for our classification task.

**Callbacks**

A screenshot of a computer

Description automatically generated

1. **ReduceLROnPlateau:**​

* Monitors validation loss.​
* Adjusts learning rate when improvement plateaus.​
* Aims to enhance convergence during training.​

1. **ModelCheckpoint:**​
   * Monitors validation loss.​
   * Saves the model at the end of each epoch.​
   * Ensures only the best-performing model is saved.​
2. **EarlyStopping:**​

* Monitors validation loss.​
* Stops training when improvement stalls.​
* Prevents overfitting by ending training at an optimal point.

**Optimizer and Loss function**

A screenshot of a computer program

Description automatically generated

* **Optimizer (Adagrad):**​
  + Adapts learning rates for each parameter.​
  + Helps converge faster in different directions.​
* **Loss Function (Categorical Crossentropy):**​
  + Measures the difference between predicted and actual distributions.​
  + Suitable for multi-class classification tasks.​
* **Metrics (Accuracy):**​
  + Evaluates model performance during training.​
  + Represents the proportion of correctly classified instances.

**MODEL SUMMARY**

A screenshot of a computer code

Description automatically generated

**TRAINING THE MODEL**

A screenshot of a computer program

Description automatically generated

**PROJECT DISTINCTIONS AND ENHANCEMENTS**

#### **1. VGG16 Model Selection:**

#### **Context:**

* + We chose the VGG16 model for melanoma detection, inspired by the research paper "Melanoma detection by analysis of clinical images using convolutional neural network."
* **Enhancements:**
  + Added extra dense layers to the VGG16 model, extending its architecture for improved feature learning and representation.

#### **2. Image Pre-processing Strategies:**

* **Context:**
  + We incorporated techniques from the research paper "Image pre-processing in computer vision systems for melanoma detection."

#### **3. CNN-SVM Hybrid Architecture:**

* **Context:**
  + Implementation plan influenced by the research paper "An architecture combining convolutional neural network (CNN) and support Vector machine (SVM) for image classification."

#### **4. Evaluation Strategy and Model Comparison:**

* **Context:**
  + Inspired by the research paper that compared softmax and SVM on the MINIST database, we plan to apply a similar evaluation to melanoma detection.
* **Enhancements:**
  + Intending to assess and compare the accuracies of models with SVM and softmax output layers using Melanoma cancer disease data, providing insights into the most effective classification approach.

**ve**

**REFERENCES**

* Agarap, A. F. (2017, December 10). "An architecture combining convolutional neural network (CNN) and support Vector machine (SVM) for image classification" , arXiv.org.
  + **Link**: <https://arxiv.org/abs/1712.03541>
* E. Nasr-Esfahani *et al*., "Melanoma detection by analysis of clinical images using convolutional neural network," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, USA, 2016, pp. 1373-1376, doi: 10.1109/EMBC.2016.7590963
  + **Link**: <https://ieeexplore.ieee.org/abstract/document/7590963>
* E. Vocaturo, E. Zumpano and P. Veltri, "Image pre-processing in computer vision systems for melanoma detection," *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Madrid, Spain, 2018, pp. 2117-2124, doi: 10.1109/BIBM.2018.8621507
  + **Link**: <https://ieeexplore.ieee.org/abstract/document/8621507>
* Neural networks:
  + **Link**: <https://www.ibm.com/topics/convolutional-neural-networks>
* Visual Geometry Group Architecture
* **Link**: <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>
* Keras Library:
* **Link**: <https://keras.io/api/>