Neural Network Project

- The model is a sequential model, which means that the layers are arranged in a linear stack. This is a common architecture for CNNs for image classification tasks , as it allows for straightforward passing of data through the layers. Each layer has a specific function and contributes to the overall ability of the model to learn from image data.
- It is featuring a combination of convolutional layers for feature extraction and dense layers for classification. The use of max pooling and dropout helps to reduce overfitting and computational load. The 'categorical_crossentropy' loss function is appropriate for multi-class classification tasks,

1. Convolutional Layers (Conv2D):

❖ The model has three convolutional layers with filter sizes 32, 64, and 128, respectively.
❖ Each convolutional layer uses a kernel size of (3, 3) and a 'relu' activation function.
❖ These layers are responsible for extracting features from the input images by applying filters. The first layer typically learns basic features like edges and textures, while deeper layers learn more complex features.

2. Max Pooling Layers (MaxPooling2D):

   - After each convolutional layer, there's a max pooling layer with a pool size of (2, 2).
   - Max pooling reduces the spatial dimensions (height and width) of the input volume for the next convolutional layer. It helps reduce computation and controls overfitting by providing an abstracted form of the representation.

3. Flatten Layer (Flatten):

   - This layer flattens the 3D output of the last max pooling layer into a 1D array. It's necessary because fully connected layers (dense layers) expect 1D inputs.

4. Dense Layers (Dense):

❖ The model includes two dense layers with 256 and 128 units, respectively, and both use 'relu' activation.
❖ Dense layers are fully connected layers that learn non-linear combinations of the high-level features extracted by the convolutional layers.

5. Dropout Layer (Dropout):

- A dropout layer with a rate of 0.5 is used.
- This layer randomly sets a fraction of the input units to 0 at each update during training, which helps prevent overfitting.

6. Output Layer:

- The final layer is a dense layer with a number of units equal to the number of classes in your dataset, using a 'softmax' activation function.
- This layer outputs a probability distribution over the classes, where the class with the highest probability is taken as the model's prediction.

Loss function:

Categorical Crossentropy:

- The loss function used in my model is 'categorical_crossentropy'.
- It's a common choice for multi-class classification tasks where the outputs are mutually exclusive.
- It Is working well with the softmax activation in the output layer to provide a probability distribution over the classes.
- This loss function compares the predicted probability distribution with the true distribution (the one-hot encoded labels), and the goal during training is to minimize this loss.

Zeina ahmed

320210137

AID sec 2