Report on Binary Classification Model for Grapes and Raisins

Group 12: 1910576101, 1910476108, 1911176114, 1910676148

March 2, 2024

1 Introduction

The objective of this experiment is to build a Fully Connected Neural Network and Convolutional Neural Network based Binary Classifier capable of distinguishing between images of grapes and raisins. The model architecture, based on Convolutional Neural Networks (CNNs) and Fully Connected Neural Networks (FCNNs) is designed to work with grayscale images of size 32x32 pixels. The goal is to achieve high accuracy in the classification task.

2 Dataset

The dataset consists of 425 grayscale images of both grapes and raisins, each with dimensions 32x32 pixels. The dataset is balanced, with an equal number of samples for each class, preventing class imbalance issues.

The images were pre-processed by reducing the size by 32x32 pixels and converting them from colour images to grayscale images.

3 Model Architecture

Both the model architectures are designed using the Keras functional API.

3.1 Convolutional Neural Network

Three sets of Convolution layers with Max Pooling layers and lastly one set of Dense layers were used as the model architecture. The model has 301633 trainable parameters. Adam optimizer is used for parameter updates during training. Binary crossentropy is employed as the loss function, suitable for binary classification tasks. Model performance is evaluated using accuracy.

3.2 Fully Connected Neural Network

Five dense layers of units(128, 256, 512, 256, 128) at first increasing then decreasing order were used. The model has 460161 trainable parameters. Adam optimizer is used for parameter updates during training. Binary crossentropy is employed as the loss function, suitable for binary classification tasks. Model performance is evaluated using accuracy.

```python
inputs = Input(shape = (32, 32, 1))
x = Conv2D(32, (3, 3), activation = "relu")(inputs)
x = Conv2D(32, (3, 3), activation = "relu")(x)
x = MaxPooling2D((2, 2))(x)
x = Conv2D(64, (3, 3), activation = "relu")(x)
x = Conv2D(64, (3, 3), activation = "relu")(x)
x = Conv2D(64, (3, 3), activation = "relu")(x)
x = MaxPooling2D((2, 2))(x)
x = Conv2D(32, (3, 3), activation = "relu")(x)
x = MaxPooling2D((2, 2))(x)
x = Flatten()(x)
x = Dense(512, activation = "relu")(x)
x = Dense(256, activation = "relu")(x)
x = Dense(128, activation = "relu")(x)
outputs = Dense(1, activation = "sigmoid")(x)

model = Model(inputs = inputs, outputs = outputs)

model.compile(optimizer = "adam", loss = "binary_crossentropy", metrics = ["accuracy"])
model.summary()
```

Figure 1: Convolutional Neural Network Architecture

```python
inputs = Input(shape = (32, 32, 1))
x = Flatten()(inputs)

x = Dense(128, activation = "relu")(x)
x = Dense(256, activation = "relu")(x)
x = Dense(512, activation = "relu")(x)
x = Dense(256, activation = "relu")(x)
x = Dense(128, activation = "relu")(x)

outputs = Dense(1, activation = "sigmoid")(x)

model = Model(inputs = inputs, outputs = outputs)

model.compile(optimizer = Adam(0.00015), loss = "binary_crossentropy", metrics = ["accuracy"])

model.summary()
```

Figure 2: Fully Connected Neural Network Architecture

4 Result

4.1 Convolutional Neural Network

After training the model, we achieved an accuracy of 90% on the test dataset.

Learning Rate	Epochs	Test Accuracy	Test Loss
0.01	10	50.59%	69.33%
0.01	15	45.88%	69.49%
0.01	20	50.59%	69.31%
0.01	30	45.29%	69.35%
0.01	50	49.41%	69.35%
0.001	10	76.47%	50.79%
0.001	15	76.47%	70.64%
0.001	20	79.41%	74.36%
0.001	30	82.35%	52.14%
0.001	**50**	**90.59%**	**76.57%**
0.0001	10	80.0%	37.86%
0.0001	15	79.41%	46.75%
0.0001	20	87.65%	32.01%
`0.0001	30	84.71%	34.68%
0.0001	**50**	**88.24%**	**30.12%**

Figure 3: Results of different settings using CNN Model

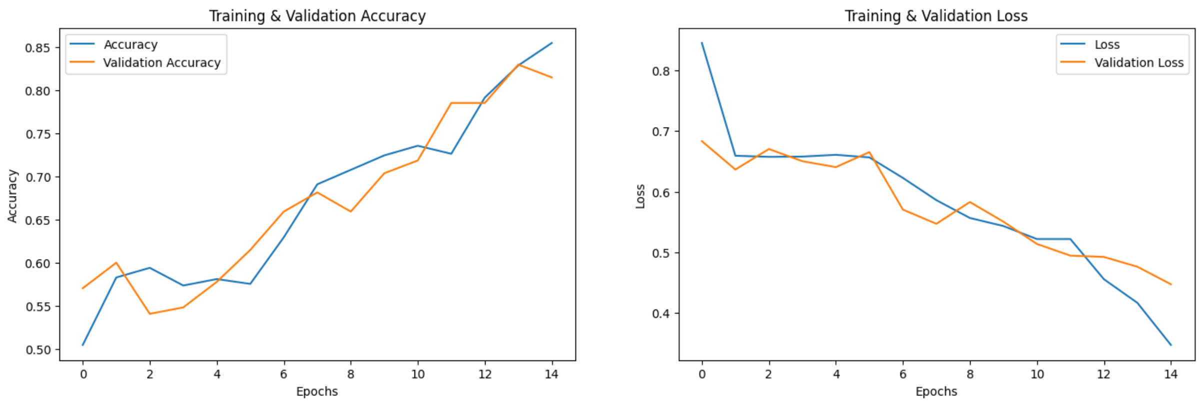

Figure 4: Accuracy and Loss Graph for CNN Model

The model seems to perform well in distinguishing between grapes and raisins in the given context.

4.2 Fully Connected Neural Network

After training the model, we achieved an accuracy of 84% on the test dataset.

Learning Rate	Epochs	Test Accuracy	Test Loss
0.0001	10	79.41%	54.51%
0.0001	15	80.01%	52.22%
0.0001	**20**	**84.71%**	**41.98%**
0.0001	30	79.41%	82.16%
0.0001	50	75.88%	74.32%
0.001	10	67.06%	66.0%
0.001	15	74.71%	54.66%
0.001	**20**	**80.0%**	**43.44%**
0.001	30	77.06%	53.62%
0.001	50	74.12%	69.68%
0.01	10	47.06%	69.46%
0.01	15	52.35%	69.2%
0.01	20	64.12%	69.75%
0.01	30	50.0%	68.15%
0.01	50	48.82%	72.22%

Figure 5: Results of different settings using FCNN Model

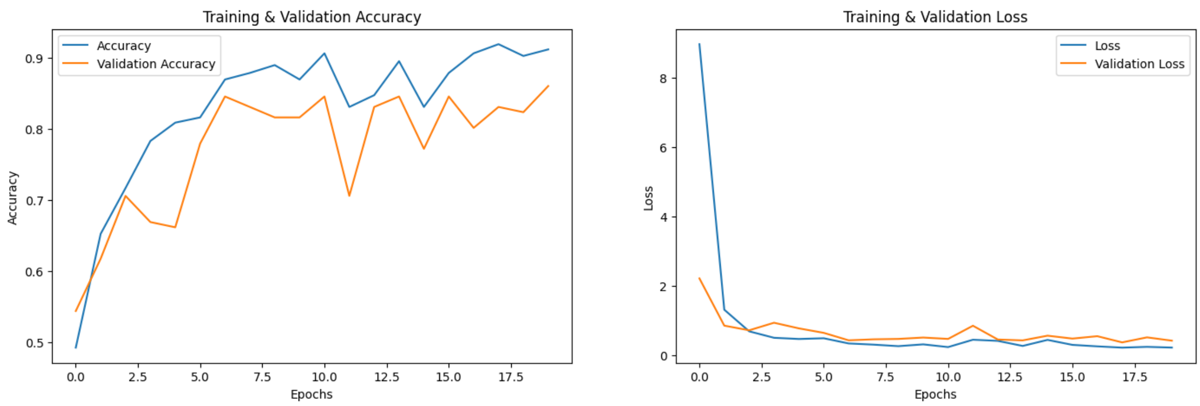

Figure 6: Accuracy and Loss Graph for FCNN Model

The model seems to perform well in distinguishing between grapes and raisins in the given context.

5 Discussion

Both models, the CNN and FCNN, demonstrated good results in classifying between grapes and raisins. While the CNN model outperformed the FCNN model slightly, both achieved accuracies above 84%. Both model's architectures were built after many trials and errors. We experimented with both models with different learning rates and epochs. Overall the experiment was successful. If the dataset was larger, we could obtain a better result.