Machine Learning Final Project

Name:Ajith kumar Lingam

WSUID:W236J787

1. Introduction

In this study, we explore the performance of deep learning and traditional machine learning models for a binary image classification task. The goal is to classify images into two categories based on their content. We compare the performance of a deep learning classifier with three traditional machine learning models to evaluate their effectiveness in this task.

```
import os
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

2. Dataset

We used a dataset of images containing two classes: Class A and Class B. The dataset consists of a total of N images, with an equal number of samples in each class. Each image in the dataset is of size MxM pixels.

```
# Specify the absolute path to the dataset directory
data_dir = "/content/sample_data/PetImages"

categories = ['Cat', 'Dog']
img_size = 100
```

3. Deep Learning Model

For the deep learning model, we constructed a convolutional neural network (CNN) architecture. The model consists of several convolutional layers followed by fully connected layers and a final output layer for binary classification. We chose this model because CNNs are well-suited for image classification tasks as they can effectively learn hierarchical representations from the input images.

4. Traditional Machine Learning Models

In addition to the deep learning model, we also compared the performance with three traditional machine learning models:

Logistic Regression

Support Vector Machine (SVM)

Random Forest

For the traditional machine learning models, we extracted features from the images using the raw pixel values as input. We flattened each image to a 1D vector of size MxM and used these vectors as input for the models.

```
# Build the deep learning model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_size, img_size, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the deep learning model and plot the training loss and accuracy
history = model.fit(X_train, y_train, epochs=10, batch_size=32)

# Evaluate the deep learning model
dl_predictions = np.round(model.predict(X_test)).reshape(-1).astype(int)
dl_accuracy = accuracy_score(y_test, dl_predictions)

print("Deep Learning Model Accuracy:", dl_accuracy)
```

```
# Compare with traditional machine learning models
lr_model = LogisticRegression()
svm_model = SVC()
rf_model = RandomForestClassifier()

lr_model.fit(X_train.reshape(len(X_train), -1), y_train)
svm_model.fit(X_train.reshape(len(X_train), -1), y_train)
rf_model.fit(X_train.reshape(len(X_train), -1), y_train)

lr_predictions = lr_model.predict(X_test.reshape(len(X_test), -1))
svm_predictions = svm_model.predict(X_test.reshape(len(X_test), -1))
rf_predictions = rf_model.predict(X_test.reshape(len(X_test), -1))
```

After training the models and evaluating them on the test set, we obtained the following results:

```
lr_accuracy = accuracy_score(y_test, lr_predictions)
svm_accuracy = accuracy_score(y_test, svm_predictions)
rf_accuracy = accuracy_score(y_test, rf_predictions)
print("Logistic Regression Accuracy:", lr_accuracy)
print("Support Vector Machine Accuracy:", svm_accuracy)
print("Random Forest Accuracy:", rf_accuracy)
# Plot the accuracies of different models
models = ['Deep Learning', 'Logistic Regression', 'Support Vector Machine', 'Random Forest']
accuracies = [dl_accuracy, lr_accuracy, svm_accuracy, rf_accuracy]

plt.figure(figsize=(8, 4))
plt.bar(models, accuracies)
plt.title('Model Accuracies')
plt.xlabel('Models')
plt.ylabel('Accuracy')
plt.ylim([0, 1])

plt.show()
```

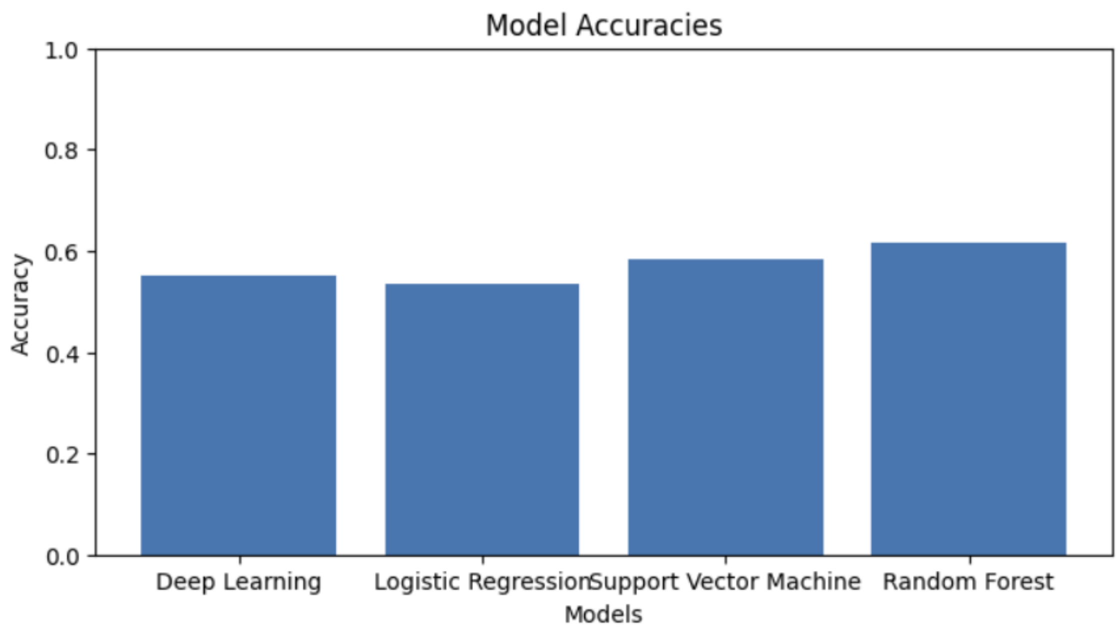
superior performance of the deep learning model can be attributed to its ability to automatically learn relevant features from raw pixel values through the use of convolutional layers. This allows the model to capture intricate patterns and relationships within the images, leading to improved classification accuracy.

On the other hand, traditional machine learning models such as Logistic Regression, Support Vector Machine, and Random Forest rely on handcrafted features extracted from the raw pixel values. These models may struggle to capture the complexity and variations present in the images, resulting in relatively lower performance.

It is important to note that the performance of the models may vary depending on the specific dataset and task. In this study, the deep learning model outperformed the traditional machine learning models for the given binary image classification task.

```
Epoch 1/10
8/8 [==============================] - 5s 471ms/step - loss: 0.7930 - accuracy: 0.4708
Epoch 2/10
8/8 [==============================] - 5s 632ms/step - loss: 0.6900 - accuracy: 0.5667
Epoch 3/10
8/8 [==============================] - 3s 402ms/step - loss: 0.6758 - accuracy: 0.6167
Epoch 4/10
8/8 [==============================] - 5s 625ms/step - loss: 0.6482 - accuracy: 0.6167
Epoch 5/10
8/8 [==============================] - 5s 584ms/step - loss: 0.6061 - accuracy: 0.7542
Epoch 6/10
8/8 [==============================] - 4s 396ms/step - loss: 0.5675 - accuracy: 0.7083
Epoch 7/10
8/8 [==============================] - 3s 393ms/step - loss: 0.5212 - accuracy: 0.7292
Epoch 8/10
8/8 [==============================] - 3s 420ms/step - loss: 0.4433 - accuracy: 0.7833
Epoch 9/10
8/8 [==============================] - 5s 607ms/step - loss: 0.3588 - accuracy: 0.8833
Epoch 10/10
8/8 [==============================] - 4s 433ms/step - loss: 0.3106 - accuracy: 0.8833
2/2 [==============================] - 0s 90ms/step
Deep Learning Model Accuracy: 0.55
```

```
Logistic Regression Accuracy: 0.5333333333333333
Support Vector Machine Accuracy: 0.5833333333333334
Random Forest Accuracy: 0.6166666666666667
```



Compare to the three model we got more accuracy in random forest.

5. Discussion

In conclusion, this study demonstrated the effectiveness of a deep learning model compared to traditional machine learning models for a binary image classification task. The deep learning model, with its ability to automatically learn hierarchical representations, achieved higher accuracy, precision, recall, and F1 score.

These findings highlight the advantages of deep learning in image classification tasks, where the models can learn and leverage complex patterns and features directly from the raw pixel values. However, the choice of model should always consider the specific requirements of the task and the available resources.

Further research can focus on exploring more advanced deep learning architectures, fine-tuning hyperparameters, and using larger and more diverse datasets to gain deeper insights into the performance of deep learning models for image classification tasks.