Project Title: Dual Output Classification Model

Client Name: Subhan
Developer Name: Ahmad Ali Rafique
Date: 25/11/2024

---

1. Overview

This project involves training a deep learning model to classify images into two categories:

1. **Student Classification (Correct/Wrong Pose)**
2. **Uniform Classification (Correct/Wrong Uniform)**

3. **Batch card detection (Wear/Not Wear)**

The model is designed to accurately predict both outputs simultaneously, achieving exceptional performance metrics on the provided dataset.

---

2. Dataset Details

The dataset provided by the client was organized as follows:

- **Main Folder:** `Subhan Model Data` • **Subfolders (Students):**
  - `Amna Abdulaziz Alshehhi`
  - `Dana Waleed Alzarooni`
  - `Shouq Ahmed Alzarooni`
- **Inside Each Subfolder:**
  - `correct` folder (images of correct poses/uniforms)
  - `wrong` folder (images of incorrect poses/uniforms)

Preprocessing Steps:

1. All images were resized to **224x224 pixels** for compatibility with the deep learning model.
2. Data augmentation techniques were applied:
   - Random flips
   - Rotations
   - Normalization
3. The dataset was split into: **80% for Training** **20% for Testing**

3. Model Architecture

The model used is a **Convolutional Neural Network (CNN)** with a dual-output structure to predict:

- **Student Classification Output**
- **Uniform Classification Output**

Architecture Details:

- **Backbone Model:** Pre-trained on `ImageNet` (e.g., ResNet50).
- **Custom Layers:**
  - **Shared Layers:** Extracted features from images using convolutional and pooling layers.
  - **Student Classification Head:**
    - Fully connected layers with softmax activation for binary classification.
  - **Uniform Classification Head:**
    - Fully connected layers with softmax activation for binary classification.

Optimization:

- **Loss Function:** Categorical Crossentropy (for both outputs)
- **Optimizer:** Adam
- **Learning Rate:** 0.001

Evaluation Metrics:

- Accuracy for both outputs.

---

4. Performance Metrics

The model was evaluated on the test dataset. Below are the results:

- **Total Loss:** 0.832
- **Student Output:**
  - Loss: 0.346
  - Accuracy: **98.9%**
- **Uniform Output:**

- Loss: 0.481
- Accuracy: **93.6%**

These metrics indicate that the model performs exceptionally well and is highly reliable for the given task.

---

5. Usage Instructions

A. Integration

The trained model is saved in the format compatible with Python scripts. Follow these steps to integrate it into the provided Python script:

1. **Load the Model:**

```
python Copy
code
from tensorflow.keras.models import load_model

model = load_model('model_name.h5')
```

2. **Input Images:**
   - Ensure input images are resized to **224x224 pixels** and normalized.
3. **Prediction:** Use the following code to get predictions:

```
python Copy
code
predictions = model.predict(image_array) student_output
= predictions['student_output'] uniform_output =
predictions['uniform_output']
```

B. Dependencies

Install the following dependencies to use the model:

- **Python Version:** 3.8+
- **Libraries:**
  - TensorFlow
  - NumPy
  - OpenCV
  - Matplotlib

Install using:

```
bash
Copy
code
pip install tensorflow numpy opencv-python matplotlib
```

---

9. Deliverables

1. Trained Model: `dual_output_model.h5`
2. Python Script for Testing
3. Dataset Preprocessing Script

---

10. Contact Information

If you have any questions or need further assistance, feel free to contact me:

- **Email:** arsbussiness786@gmail.com
- **Phone:** 0328-9460713
- **GitHub Portfolio:** https://github.com/Ahmad-Ali-Rafique