UHAR_Using_Arabic_pretrained_models

October 4, 2025

1 Performance Evaluation of Swin_tiny and Deit_tiny Pre-trained on Arabic alphabet dataset

pre-trained on imagenet

Author: Zaryab rahman

Date: 4/10/25

1.0.1 Imports

[1]:
```
import os
import time
import json
import logging
import torch
import timm
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from torch import nn
from torch.utils.data import DataLoader, SubsetRandomSampler
from torchvision import datasets, transforms
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score
from tqdm.auto import tqdm
```

[ ]:
```
# unzip a file
!unzip "/content/archive (4).zip" -d "/content/uhar/"
```

1.0.2 Config

[11]:
```
CONFIG = {
    "data_path": "/content/uhar/data/data",

    "num_classes": 43,
    "batch_size": 64,
    "image_size": 224,
    "num_epochs": 50,
    "learning_rate": 0.0001, # the only hyperparam we change for this fine␣
 ↪tuning because when fine tuning this swin model on arabic we got severe␣
 ↪model collapse with higer lr it wasted our hrs of woek by chekcing and␣
 ↪verifying the data loaders which werent the issue
    "early_stopping_patience": 5,
    "validation_split": 0.15,               # 15% of training data used for␣
 ↪validation

    "results_dir": "/content/drive/MyDrive/uhar/results/",
    "checkpoints_dir": "/content/drive/MyDrive/uhar/checkpoints/",

    "models_to_evaluate": [
        # cnn
        "resnet18",
        "resnet50",
        # "vgg16", much larger so avoid for now
        "mobilenetv2_100",
        "efficientnet_b0",

        # vsion transformer
        "vit_tiny_patch16_224",
        "swin_tiny_patch4_window7_224",
        "deit_tiny_distilled_patch16_224",
        "mobilevit_s"
    ]
}
```

1.0.3 setup device, logging, and directories

[4]:
```
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

os.makedirs(CONFIG["results_dir"], exist_ok=True)
os.makedirs(CONFIG["checkpoints_dir"], exist_ok=True)

logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s - %(levelname)s - %(message)s',
                    handlers=[
                        logging.FileHandler("experiment_log.log"),
```

```
                        logging.StreamHandler()
                    ])

logging.info(f"Using device: {DEVICE}")
```

1.0.4 EarlyStopping

[5]:
```
class EarlyStopping:
    def __init__(self, patience=5, min_delta=0, checkpoint_path='checkpoint.
↪pth'):
        self.patience = patience
        self.min_delta = min_delta
        self.counter = 0
        self.best_score = None
        self.early_stop = False
        self.val_loss_min = np.inf
        self.checkpoint_path = checkpoint_path

    def __call__(self, val_loss, model):
        score = -val_loss
        if self.best_score is None:
            self.best_score = score
            self.save_checkpoint(val_loss, model)
        elif score < self.best_score + self.min_delta:
            self.counter += 1
            logging.info(f'earlyStopping counter: {self.counter} out of {self.
↪patience}')
            if self.counter >= self.patience:
                self.early_stop = True
        else:
            self.best_score = score
            self.save_checkpoint(val_loss, model)
            self.counter = 0

    def save_checkpoint(self, val_loss, model):
        logging.info(f'validation loss decreased ({self.val_loss_min:.6f} -->␣
↪{val_loss:.6f}). Saving best model...')
        torch.save(model.state_dict(), self.checkpoint_path)
        self.val_loss_min = val_loss
```

1.0.5 Data Loading and Verification

[6]:
```
def create_dataloaders(config):
    data_path = config["data_path"]
    if not os.path.exists(data_path):
        logging.error(f"Path not exists : {data_path}")
```

```
        return None, None, None, []

    data_transform = transforms.Compose([
        transforms.Resize((config["image_size"], config["image_size"])),
        transforms.Grayscale(num_output_channels=3),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.
↪225])
    ])

    train_dir = "/content/uhar/data/data/characters_train_set"
    test_dir = "/content/uhar/data/data/characters_test_set"

    if not os.path.exists(train_dir) or not os.path.exists(test_dir):
        logging.error("Training or test directory not found.")
        return None, None, None, []


    full_train_dataset = datasets.ImageFolder(train_dir,␣
↪transform=data_transform)

    # case incentivate version since there are some inconsities in the naming␣
↪of the folders

    class_to_idx_lower = {cls.lower(): idx for cls, idx in full_train_dataset.
↪class_to_idx.items()}
    class_names = full_train_dataset.classes

    config["num_classes"] = len(class_names)
    logging.info(f"Established class mapping from training data. Found␣
↪{len(class_names)} classes.")

    test_dataset = datasets.ImageFolder(test_dir, transform=data_transform)
    logging.info(f"Initially loaded test set with {len(test_dataset)} images.")

    remapped_targets = []
    valid_imgs = []

    for path, original_idx in test_dataset.imgs:
        class_name_from_folder = os.path.basename(os.path.dirname(path)).lower()

        correct_idx = class_to_idx_lower.get(class_name_from_folder)

        if correct_idx is not None:
            remapped_targets.append(correct_idx)
            valid_imgs.append((path, correct_idx))
        else:
```

```
            logging.warning(f"Class '{os.path.basename(os.path.dirname(path))}'␣
↪from test set has no equivalent in training set. Skipping image: {path}")

    test_dataset.imgs = valid_imgs
    test_dataset.samples = valid_imgs
    test_dataset.targets = remapped_targets
    test_dataset.classes = full_train_dataset.classes
    test_dataset.class_to_idx = full_train_dataset.class_to_idx

    logging.info(f"Successfully remapped test dataset. Total valid test images:␣
↪{len(test_dataset)}")


    val_split = config["validation_split"]
    dataset_size = len(full_train_dataset)
    indices = list(range(dataset_size))
    split = int(np.floor(val_split * dataset_size))
    np.random.seed(42)
    np.random.shuffle(indices)
    train_indices, val_indices = indices[split:], indices[:split]

    train_sampler = SubsetRandomSampler(train_indices)
    val_sampler = SubsetRandomSampler(val_indices)

    num_workers = 1
    train_loader = DataLoader(full_train_dataset,␣
↪batch_size=config["batch_size"], sampler=train_sampler,␣
↪num_workers=num_workers)
    val_loader = DataLoader(full_train_dataset,␣
↪batch_size=config["batch_size"], sampler=val_sampler,␣
↪num_workers=num_workers)
    test_loader = DataLoader(test_dataset, batch_size=config["batch_size"],␣
↪shuffle=False, num_workers=num_workers)

    logging.info(f"Data loaders created: {len(train_indices)} train images,␣
↪{len(val_indices)} validation images, {len(test_dataset)} test images.")

    return train_loader, val_loader, test_loader, class_names

train_loader, val_loader, test_loader, class_names = create_dataloaders(CONFIG)
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (1).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (10).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (100).jpg
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (101).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (102).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (103).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (104).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (105).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (106).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (107).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (108).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (109).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (11).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (110).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (111).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (112).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (113).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (114).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (115).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (116).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (117).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (118).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (119).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (12).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (120).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (121).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (122).jpg
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (123).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (124).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (125).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (126).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (127).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (128).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (129).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (13).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (130).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (131).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (132).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (133).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (14).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (15).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (16).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (17).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (18).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (19).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (2).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (20).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (21).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (22).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (23).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (24).jpg
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (25).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (26).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (27).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (28).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (29).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (3).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (30).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (31).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (32).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (33).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (34).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (35).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (36).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (37).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (38).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (39).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (4).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (40).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (41).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (42).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (43).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (44).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (45).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (46).jpg
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (47).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (48).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (49).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (5).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (50).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (51).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (52).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (53).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (54).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (55).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (56).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (57).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (58).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (59).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (6).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (60).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (61).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (62).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (63).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (64).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (65).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (66).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (67).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (68).jpg
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (69).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (7).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (70).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (71).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (72).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (73).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (74).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (75).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (76).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (77).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (78).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (79).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (8).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (80).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (81).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (82).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (83).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (84).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (85).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (86).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (87).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (88).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (89).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (9).jpg
```

```
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (90).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (91).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (92).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (93).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (94).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (95).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (96).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (97).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (98).jpg
WARNING:root:Class 'Twaa' from test set has no equivalent in training set.
Skipping image: /content/uhar/data/data/characters_test_set/Twaa/twaa (99).jpg
```

1.0.6 Core Training and Evaluation Functions

[7]:
```python
def train_model(model, model_name, config, train_dataloader, val_dataloader):
    loss_fn = nn.CrossEntropyLoss()
    optimizer = torch.optim.Adam(model.parameters(), lr=config["learning_rate"])
    history = {"train_loss": [], "train_acc": [], "val_loss": [], "val_acc": []}

    best_model_path = os.path.join(config["checkpoints_dir"],␣
↪f"{model_name}_best_model.pth")
    early_stopper = EarlyStopping(patience=config["early_stopping_patience"],␣
↪checkpoint_path=best_model_path)

    start_time = time.time()
    for epoch in range(config["num_epochs"]):
        model.train()
        train_loss, train_acc = 0, 0
        train_pbar = tqdm(train_dataloader, desc=f"Epoch {epoch+1} [Train]",␣
↪leave=False)
        for X, y in train_pbar:
            X, y = X.to(DEVICE), y.to(DEVICE)
            y_pred = model(X)
            loss = loss_fn(y_pred, y)
            train_loss += loss.item()
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
```

```
            y_pred_class = torch.argmax(torch.softmax(y_pred, dim=1), dim=1)
            batch_acc = (y_pred_class == y).sum().item() / len(y_pred)
            train_acc += batch_acc
            train_pbar.set_postfix(loss=loss.item(), acc=batch_acc)
        train_loss /= len(train_dataloader)
        train_acc /= len(train_dataloader)

        model.eval()
        val_loss, val_acc = 0, 0
        val_pbar = tqdm(val_dataloader, desc=f"Epoch {epoch+1} [Val]",␣
↪leave=False)
        with torch.no_grad():
            for X, y in val_pbar:
                X, y = X.to(DEVICE), y.to(DEVICE)
                y_pred = model(X)
                loss = loss_fn(y_pred, y)
                val_loss += loss.item()
                y_pred_class = torch.argmax(torch.softmax(y_pred, dim=1), dim=1)
                batch_acc = (y_pred_class == y).sum().item() / len(y_pred)
                val_acc += batch_acc
                val_pbar.set_postfix(loss=loss.item(), acc=batch_acc)
        val_loss /= len(val_dataloader)
        val_acc /= len(val_dataloader)

        logging.info(f"Epoch: {epoch+1} | Train Loss: {train_loss:.4f} | Train␣
↪Acc: {train_acc:.4f} | Val Loss: {val_loss:.4f} | Val Acc: {val_acc:.4f}")
        history["train_loss"].append(train_loss)
        history["train_acc"].append(train_acc)
        history["val_loss"].append(val_loss)
        history["val_acc"].append(val_acc)

        early_stopper(val_loss, model)
        if early_stopper.early_stop:
            logging.info("Early stopping triggered.")
            break

    train_time = time.time() - start_time
    logging.info(f"Loading best model weights from epoch checkpoint.")
    model.load_state_dict(torch.load(best_model_path))
    return model, history, train_time

def evaluate_on_test_set(model, test_dataloader, class_names):
    model.eval()
    y_true, y_pred = [], []
    start_time = time.time()
    with torch.no_grad():
        for images, labels in tqdm(test_dataloader, desc="Testing"):
```

```
            images = images.to(DEVICE)
            outputs = model(images)
            _, predicted = torch.max(outputs.data, 1)
            y_true.extend(labels.cpu().numpy())
            y_pred.extend(predicted.cpu().numpy())

    total_time = time.time() - start_time
    inference_time_ms = (total_time / len(test_dataloader.dataset)) * 1000
    accuracy = accuracy_score(y_true, y_pred)
    report = classification_report(y_true, y_pred, target_names=class_names,␣
 ↪output_dict=True, zero_division=0)
    cm = confusion_matrix(y_true, y_pred)
    return accuracy, report, cm, inference_time_ms
```

1.0.7 Visualization Functions

[8]:
```
def plot_training_curves(history, model_name, save_dir):
    plt.style.use('seaborn-v0_8-whitegrid')
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 6))
    fig.suptitle(f'Training and Validation Metrics for {model_name}',␣
 ↪fontsize=16)

    ax1.plot(history['train_loss'], label='Train Loss')
    ax1.plot(history['val_loss'], label='Validation Loss')
    ax1.set_title('Loss Curves')
    ax1.set_xlabel('Epochs'); ax1.set_ylabel('Loss'); ax1.legend()

    ax2.plot(history['train_acc'], label='Train Accuracy')
    ax2.plot(history['val_acc'], label='Validation Accuracy')
    ax2.set_title('Accuracy Curves')
    ax2.set_xlabel('Epochs'); ax2.set_ylabel('Accuracy'); ax2.legend()

    plt.savefig(os.path.join(save_dir, f"{model_name}_training_curves.png"))
    plt.close()

def plot_confusion_matrix(cm, model_name, class_names, save_dir,␣
 ↪normalize=False):
    plt.style.use('default')
    plt.figure(figsize=(16, 14))

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        fmt, title, fname = '.2f', f'Normalized Confusion Matrix:␣
 ↪{model_name}', f"{model_name}_cm_normalized.png"
    else:
        fmt, title, fname = 'd', f'Confusion Matrix: {model_name}',␣
 ↪f"{model_name}_cm.png"
```

```
    sns.heatmap(cm, annot=False, fmt=fmt, cmap='Blues',␣
↪xticklabels=class_names, yticklabels=class_names)
    plt.title(title); plt.xlabel('Predicted'); plt.ylabel('True')
    plt.savefig(os.path.join(save_dir, fname))
    plt.close()
```

1.0.8 main experiment execution

[15]:
```
def run_and_save_single_model(model_name, config, train_loader, val_loader,␣
↪test_loader, class_names):
    custom_weights_paths = {
        "swin_tiny_patch4_window7_224": "/content/
↪swin_tiny_patch4_window7_224_best_model.pth",
        "deit_tiny_distilled_patch16_224": "/content/
↪deit_tiny_distilled_patch16_224_best_model.pth"
    }

    logging.info(f"\n{'='*60}\nStarting experiment for: {model_name}␣
↪\n{'='*60}")

    if model_name in custom_weights_paths:
        logging.info(f"Found custom Arabic pre-trained weights for {model_name}.
↪")
        model = timm.create_model(
            model_name,
            pretrained=False,
            num_classes=config["num_classes"]
        ).to(DEVICE)

        weights_path = custom_weights_paths[model_name]
        logging.info(f"Loading weights from: {weights_path}")
        pretrained_state_dict = torch.load(weights_path)

        filtered_state_dict = {
            key: value for key, value in pretrained_state_dict.items()
            if not (key.startswith('head.') or key.startswith('head_dist.'))
        }

        model.load_state_dict(filtered_state_dict, strict=False)
        logging.info("Successfully loaded feature extractor weights. The␣
↪classification head is randomly initialized.")

    else:
        logging.info(f"No custom weights found. Using ImageNet pre-trained␣
↪weights for {model_name}.")
```

```
        model = timm.create_model(
            model_name,
            pretrained=True,
            num_classes=config["num_classes"]
        ).to(DEVICE)

    num_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
    logging.info(f"Trainable parameters: {num_params / 1e6:.2f}M")

    model, history, train_time = train_model(model, model_name, config,␣
↪train_loader, val_loader)
    plot_training_curves(history, model_name, config["results_dir"])
    accuracy, report, cm, inference_time = evaluate_on_test_set(model,␣
↪test_loader, class_names)
    plot_confusion_matrix(cm, model_name, class_names, config["results_dir"],␣
↪normalize=False)
    plot_confusion_matrix(cm, model_name, class_names, config["results_dir"],␣
↪normalize=True)

    model_results = {
        "Model": model_name,
        "Type": "ViT" if any(x in model_name for x in ["vit", "swin", "deit"])␣
↪else "CNN",
        "Params (M)": round(num_params / 1e6, 2),
        "Train Time (s)": round(train_time, 2),
        "Test Accuracy": round(accuracy, 4),
        "F1-Score (Macro)": round(report["macro avg"]["f1-score"], 4),
        "Inference Time (ms/img)": round(inference_time, 2)
    }

    result_filepath = os.path.join(CONFIG["results_dir"],␣
↪f"{model_name}_results.json")
    with open(result_filepath, 'w') as f:
        json.dump(model_results, f, indent=4)

    logging.info(f"Results for {model_name} saved to {result_filepath}")
    logging.info(f"Finished experiment for: {model_name}")
```

[ ]:
```
model_to_run_now = "deit_tiny_distilled_patch16_224"

if model_to_run_now not in CONFIG["models_to_evaluate"]:
    logging.error(f"error: Model '{model_to_run_now}' not found in␣
↪CONFIG['models_to_evaluate'].")
    logging.error(f"please choose from: {CONFIG['models_to_evaluate']}")
else:
    if train_loader:
```

```
        run_and_save_single_model(model_to_run_now, CONFIG, train_loader,␣
↪val_loader, test_loader, class_names)
    else:
        logging.error("Data cant be loaded")
```

[ ]:

2 Imagenet-vs-Arabic-Pretrained final performance comparision

[17]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import json
import glob
import os

results_dir = CONFIG["results_dir"]
all_files = glob.glob(os.path.join(results_dir, "*_results.json"))

all_results = []
for f in all_files:
    with open(f, 'r') as file:
        data = json.load(file)

        if 'arabic' in f:
            data['Pre-training'] = 'Arabic Pre-trained'
        else:
            data['Pre-training'] = 'ImageNet Pre-trained'

        if 'swin_tiny' in data['Model']:
            data['Model_Short'] = 'Swin-Tiny'
        elif 'deit_tiny' in data['Model']:
            data['Model_Short'] = 'DeiT-Tiny'
        elif 'efficientnet_b0' in data['Model']:
             data['Model_Short'] = 'EfficientNet-B0'
        else:
            data['Model_Short'] = data['Model']

        all_results.append(data)

if not all_results:
    print("No result files found! Please check your file names and paths.")
else:
    summary_df = pd.DataFrame(all_results)

    print("="*80)
```

```
    print("Aggregated Performance Summary")
    print("="*80)
    display_cols = ['Model_Short', 'Pre-training', 'Test Accuracy', 'F1-Score␣
↪(Macro)', 'Train Time (s)', 'Inference Time (ms/img)']
    print(summary_df[display_cols].sort_values(by=['Model_Short',␣
↪'Pre-training']))

    plt.style.use('seaborn-v0_8-whitegrid')
    fig, axes = plt.subplots(2, 2, figsize=(22, 18))
    fig.suptitle('Performance Comparison: Arabic vs. ImageNet Pre-training on␣
↪UHAR Dataset', fontsize=24, y=0.95)

    plot_order = sorted(summary_df['Model_Short'].unique())

    sns.barplot(data=summary_df, x='Model_Short', y='Test Accuracy',␣
↪hue='Pre-training', ax=axes[0, 0], order=plot_order)
    axes[0, 0].set_title('Peak Test Accuracy', fontsize=16)
    axes[0, 0].set_xlabel('')
    axes[0, 0].set_ylabel('Accuracy (%)', fontsize=12)
    if summary_df['Test Accuracy'].min() > 0.85:
        axes[0, 0].set_ylim(0.90, summary_df['Test Accuracy'].max() + 0.01)

    sns.barplot(data=summary_df, x='Model_Short', y='F1-Score (Macro)',␣
↪hue='Pre-training', ax=axes[0, 1], order=plot_order)
    axes[0, 1].set_title('Macro F1-Score', fontsize=16)
    axes[0, 1].set_xlabel('')
    axes[0, 1].set_ylabel('F1-Score', fontsize=12)
    if summary_df['F1-Score (Macro)'].min() > 0.85:
        axes[0, 1].set_ylim(0.90, summary_df['F1-Score (Macro)'].max() + 0.01)

    sns.barplot(data=summary_df, x='Model_Short', y='Train Time (s)',␣
↪hue='Pre-training', ax=axes[1, 0], order=plot_order)
    axes[1, 0].set_title('Total Training Time', fontsize=16)
    axes[1, 0].set_xlabel('Model', fontsize=12)
    axes[1, 0].set_ylabel('Time (seconds)', fontsize=12)

    sns.barplot(data=summary_df, x='Model_Short', y='Inference Time (ms/img)',␣
↪hue='Pre-training', ax=axes[1, 1], order=plot_order)
    axes[1, 1].set_title('Inference Time per Image', fontsize=16)
    axes[1, 1].set_xlabel('Model', fontsize=12)
    axes[1, 1].set_ylabel('Time (milliseconds)', fontsize=12)

    for ax in axes.flat:
        for p in ax.patches:
            ax.annotate(f'{p.get_height():.2f}',
                        (p.get_x() + p.get_width() / 2., p.get_height()),
```

```
                        ha='center', va='center',
                        xytext=(0, 9),
                        textcoords='offset points',
                        fontsize=10)
    plt.tight_layout(rect=[0, 0.03, 1, 0.93])
    plt.savefig(os.path.join(results_dir,␣
↪"summary_comparison_arabic_vs_imagenet.png"))
    plt.show()
```

```
================================================================================
Aggregated Performance Summary
================================================================================
  Model_Short          Pre-training  Test Accuracy  F1-Score (Macro)  \
1   DeiT-Tiny    Arabic Pre-trained         0.9803            0.9603
3   DeiT-Tiny  ImageNet Pre-trained         0.9743            0.9503
2   Swin-Tiny    Arabic Pre-trained         0.9888            0.9642
0   Swin-Tiny  ImageNet Pre-trained         0.9216            0.9000

   Train Time (s)  Inference Time (ms/img)
1         2680.62                    3.05
3         2680.62                    3.05
2         3327.31                    4.47
0         5586.76                    4.65
```