یک مدل جدیدتر برای مسأله تقسیم‌بندی تصاویر، مدل \*\*Swin UNet\*\* است که مبتنی بر معماری ترانسفورمرهای ویژوال است. این مدل که در سال‌های اخیر معرفی شده، ترکیبی از \*\*Swin Transformer\*\* و U-Net را به کار می‌گیرد و برای تقسیم‌بندی تصاویر کاربرد گسترده‌ای دارد.

\*\*Swin Transformer\*\* (که خود یک مدل بسیار جدیدتر نسبت به معماری‌های کلاسیک مثل CNNهاست) از مکانیزم توجه (Attention Mechanism) به جای لایه‌های معمول کانولوشنی برای پردازش تصاویر استفاده می‌کند و در این مدل‌ها پویایی و دقت بیشتری در تشخیص نواحی مختلف تصویر حاصل می‌شود.

### پیاده‌سازی مدل Swin UNet:

اینجا یک نمونه ساده از معماری Swin UNet آورده شده است که برای تقسیم‌بندی تصویر مناسب است:

```python

import tensorflow as tf

from tensorflow.keras import layers, Model

# Swin Transformer Block

def swin\_transformer\_block(x, num\_heads, key\_dim, window\_size):

# Multi-Head Self Attention (Window-based)

x = layers.MultiHeadAttention(num\_heads=num\_heads, key\_dim=key\_dim)(x, x)

x = layers.LayerNormalization()(x)

# Feed-forward Network (MLP)

x = layers.Dense(units=x.shape[-1], activation='relu')(x)

x = layers.LayerNormalization()(x)

return x

# Swin-UNet Model

def swin\_unet(input\_size=(256, 256, 3)):

inputs = tf.keras.Input(input\_size)

# Patch Embedding

x = layers.Conv2D(64, (4, 4), strides=(4, 4), padding="same")(inputs)

# Swin Transformer Layers

for \_ in range(2):

x = swin\_transformer\_block(x, num\_heads=4, key\_dim=64, window\_size=7)

# Decoder (Upsampling with Skip Connections)

x = layers.Conv2DTranspose(64, (2, 2), strides=(2, 2), padding="same")(x)

outputs = layers.Conv2D(1, (1, 1), activation='sigmoid')(x)

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

return model

model = swin\_unet(input\_size=(256, 256, 3))

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.summary()

```

### مقاله‌های مرتبط با Swin UNet:

1. [Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation](https://arxiv.org/abs/2105.05537) — این مقاله نشان می‌دهد که استفاده از مکانیزم ترانسفورمر در کنار معماری U-Net باعث بهبود عملکرد مدل‌های بخش‌بندی تصویر می‌شود.

2. [Swin Transformer: Hierarchical Vision Transformer using Shifted Windows](https://arxiv.org/abs/2103.14030) — مدل Swin Transformer به عنوان پایه‌ای برای Swin-UNet عمل می‌کند و در این مقاله معرفی شده است.

استفاده از این مدل‌ها در مقایسه با مدل‌های کلاسیک مانند U-Net، به دلیل قدرت ترانسفورمرها در یادگیری بهتر ویژگی‌های پیچیده تصویر، منجر به بهبود دقت تقسیم‌بندی می‌شود.

یک مدل جدیدتر برای مسأله تقسیم‌بندی تصاویر، مدل \*\*Swin UNet\*\* است که مبتنی بر معماری ترانسفورمرهای ویژوال است. این مدل که در سال‌های اخیر معرفی شده، ترکیبی از \*\*Swin Transformer\*\* و U-Net را به کار می‌گیرد و برای تقسیم‌بندی تصاویر کاربرد گسترده‌ای دارد.

\*\*Swin Transformer\*\* (که خود یک مدل بسیار جدیدتر نسبت به معماری‌های کلاسیک مثل CNNهاست) از مکانیزم توجه (Attention Mechanism) به جای لایه‌های معمول کانولوشنی برای پردازش تصاویر استفاده می‌کند و در این مدل‌ها پویایی و دقت بیشتری در تشخیص نواحی مختلف تصویر حاصل می‌شود.

### پیاده‌سازی مدل Swin UNet:

اینجا یک نمونه ساده از معماری Swin UNet آورده شده است که برای تقسیم‌بندی تصویر مناسب است:

### مقاله‌های مرتبط با Swin UNet:

1. [Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation](https://arxiv.org/abs/2105.05537) — این مقاله نشان می‌دهد که استفاده از مکانیزم ترانسفورمر در کنار معماری U-Net باعث بهبود عملکرد مدل‌های بخش‌بندی تصویر می‌شود.

2. [Swin Transformer: Hierarchical Vision Transformer using Shifted Windows](https://arxiv.org/abs/2103.14030) — مدل Swin Transformer به عنوان پایه‌ای برای Swin-UNet عمل می‌کند و در این مقاله معرفی شده است.

استفاده از این مدل‌ها در مقایسه با مدل‌های کلاسیک مانند U-Net، به دلیل قدرت ترانسفورمرها در یادگیری بهتر ویژگی‌های پیچیده تصویر، منجر به بهبود دقت تقسیم‌بندی می‌شود.

-----------You can access the full paper for "Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation" on [arXiv](https://ar5iv.labs.arxiv.org/html/2105.05537)【30†source】. This paper presents a model that combines U-Net's architecture with Swin Transformers, designed for medical image segmentation tasks like cardiac and multi-organ segmentation. You can also find code and further resources related to the project on the authors' [GitHub repository](https://github.com/HuCaoFighting/Swin-Unet)【29†source】.

Here’s a `README.md` file for your GitHub repository documenting the U-Net and Swin Transformer UNet models for image segmentation.

---

# Image Segmentation using U-Net and Swin Transformer UNet

This repository contains two deep learning models for medical image segmentation: \*\*U-Net\*\* and \*\*Swin Transformer UNet\*\*. Both models are implemented using TensorFlow/Keras and are designed to perform pixel-wise classification, often used for tasks like medical imaging, autonomous driving, or any other segmentation-related problems.

## Table of Contents

- [Introduction](#introduction)

- [Models](#models)

- [U-Net Model](#u-net-model)

- [Swin Transformer UNet](#swin-transformer-unet)

- [Data Augmentation](#data-augmentation)

- [Usage](#usage)

- [Requirements](#requirements)

- [Training](#training)

- [Evaluation](#evaluation)

- [References](#references)

## Introduction

Image segmentation is crucial in various fields, such as medical imaging and autonomous driving. The two models implemented here—\*\*U-Net\*\* and \*\*Swin Transformer UNet\*\*—offer powerful solutions for segmentation tasks:

- \*\*U-Net\*\*: A convolutional neural network architecture that uses an encoder-decoder structure with skip connections for efficient feature extraction and image reconstruction.

- \*\*Swin Transformer UNet\*\*: A transformer-based architecture, utilizing the Swin Transformer as the backbone. It provides excellent long-range context understanding due to its self-attention mechanism.

## Models

### U-Net Model

The U-Net model is based on the popular encoder-decoder structure, which consists of a contracting path (downsampling) followed by an expansive path (upsampling). The key feature of U-Net is its \*\*skip connections\*\*, which help preserve high-resolution features and improve the model's segmentation capabilities.

![U-Net Architecture](unet.png)

Key components:

- \*\*Encoder\*\*: Extracts features from the image using a series of convolutional layers followed by max-pooling.

- \*\*Bottleneck\*\*: Contains the highest-level features before upsampling begins.

- \*\*Decoder\*\*: Upsamples the features to match the original input resolution.

- \*\*Skip Connections\*\*: Directly connect encoder layers to decoder layers to retain spatial information.

### Swin Transformer UNet

The \*\*Swin Transformer UNet\*\* utilizes the Swin Transformer, which is based on self-attention mechanisms. Swin Transformers capture long-range dependencies and have strong performance on various vision tasks, including image segmentation. In the Swin Transformer UNet, the image is processed in windows of patches and features are extracted using attention layers.

Key components:

- \*\*Self-Attention\*\*: Captures global information in images through attention mechanisms.

- \*\*Patch-Based Processing\*\*: The image is divided into patches, and transformations are applied to capture both local and global features.

- \*\*Decoder\*\*: Similar to U-Net, but it integrates the transformer layers.

## Data Augmentation

The training dataset undergoes several augmentations to improve the model's generalization capabilities. The augmentations include:

- \*\*Flipping\*\*: Random horizontal/vertical flips.

- \*\*Rotation\*\*: Random rotations within a specified range.

- \*\*Scaling\*\*: Random scaling of the image.

- \*\*Translation\*\*: Random pixel shifts.

- \*\*Noise Injection\*\*: Adding random noise to the image for robustness.

These transformations are applied using a custom function with the `tf.raw\_ops.ImageProjectiveTransformV2` operation.

## Usage

### Requirements

To run the models, you need to have the following libraries installed:

- TensorFlow (>= 2.0)

- Keras

- Matplotlib

- Python 3.x

Install the required packages with:

```bash

pip install tensorflow keras matplotlib

```

### Training

To train the models, the dataset should be structured into two folders:

- \*\*frames/\*\*: Contains the input images.

- \*\*masks/\*\*: Contains the corresponding binary masks for each image.

Run the script to train either the U-Net or Swin Transformer UNet model:

```bash

python train.py --model unet --data\_dir /path/to/data/

```

For Swin Transformer UNet:

```bash

python train.py --model swin --data\_dir /path/to/data/

```

### Evaluation

The model can be evaluated using a separate script. After training, evaluate the model on a validation dataset:

```bash

python evaluate.py --model /path/to/saved\_model --data\_dir /path/to/validation\_data/

```

The model's performance is measured using:

- \*\*Dice Coefficient\*\*: A common evaluation metric for image segmentation.

- \*\*Accuracy\*\*: Pixel-wise classification accuracy.

## References

- \*\*U-Net Paper\*\*: [U-Net: Convolutional Networks for Biomedical Image Segmentation](https://arxiv.org/abs/1505.04597)

- \*\*Swin Transformer Paper\*\*: [Swin Transformer: Hierarchical Vision Transformer using Shifted Windows](https://arxiv.org/abs/2103.14030)

Feel free to explore the repository, and raise issues if you encounter any problems!

---

This should work as a detailed yet concise readme for your project.