RRU-Net

The Ringed Residual U-Net for Image Splicing Forgery Detection

Gioele Migno
1795826

Aim of the project

The objective of the project is to develop a neural network capable of detecting a splicing forgery image, in particular locating the specific forgery region.

The splicing forgery is a image manipulation technique that copies parts of one image and then pastes it into another one to obtain a new image which usually has a different meaning.

Existing methods

Since the tampered regions come from other images, there are several differences between the un-tampered and tampered parts, such as lighting and sensor noise.

According to the feature extraction methods used in the existing splicing forgery detection method, they can be classified into two main categories:

- **Traditional features extraction methods**(e.g. based on the imaging device attribute or image compression attribute)
- **CNN-based detection methods**

Existing methods: **Current Issues**

- **Traditional features extraction methods**: They can be circumvented by post-processing techniques like JPEG compression and noise corruption
- **CNN-based detection methods:** Since they use image paths as input, the contextual spatial information is lost, moreover, when the network is deeper, the gradient degradation makes discrimination of features weaker

Ringed Residual U-Net

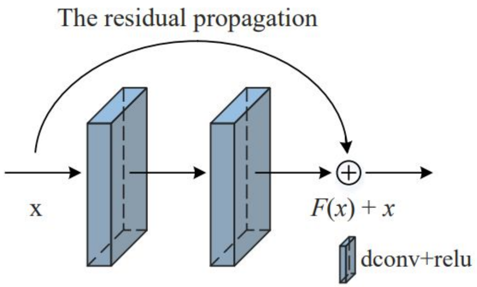
U-Net architecture is widely used for image segmentation task due to its capability to capture context information thanks to the downsampling, upsampling layers and the connection between different resolutions layers.

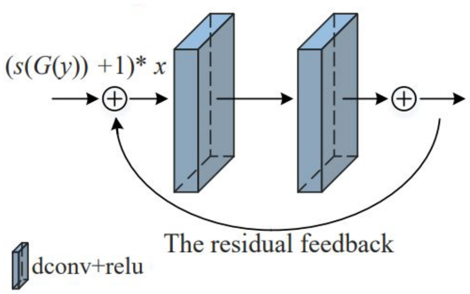
Since image splicing detection can be viewed as a complicated image segmentation task, independent of the human visual system, the authors proposed to modify U-Net architecture:

- Residual Propagation
- Residual Feedback

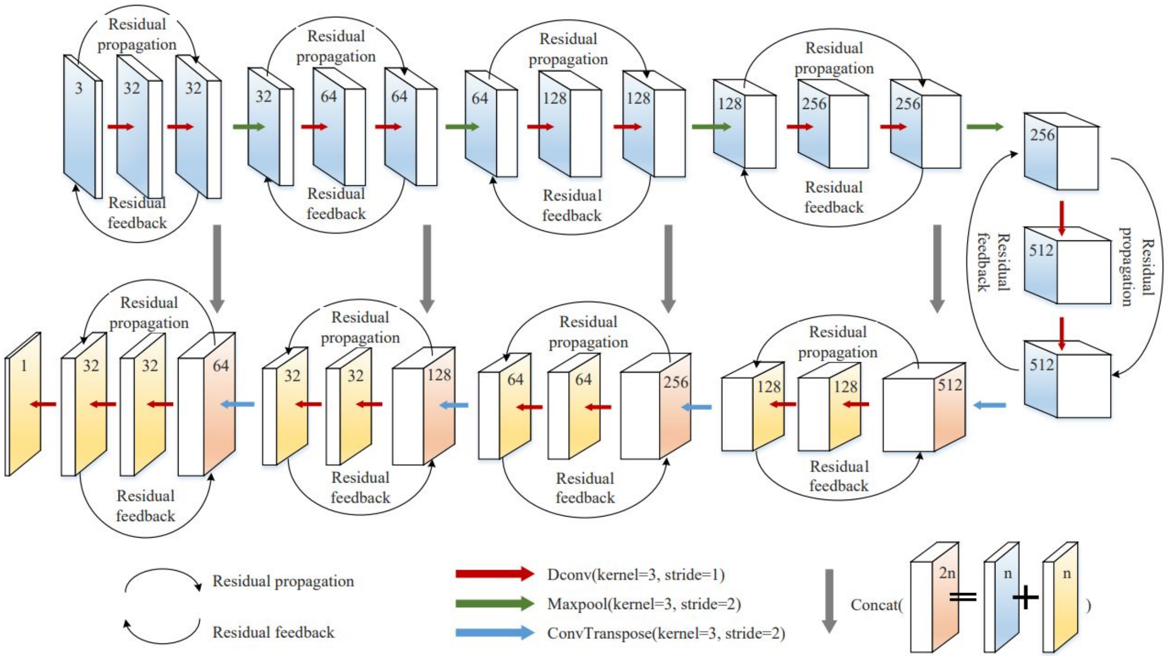
Ringed Residual U-Net: **U-Net Changes**

Residual Propagation	Residual Feedback
- Inspired by **recall** in human brain - Solves gradient degradation problem	- Inspired by **consolidation** in human brain - Amplifies differences between tampered and un-tampered regions





Ringed Residual U-Net: **Overall Architecture**



Datasets

Original paper used:

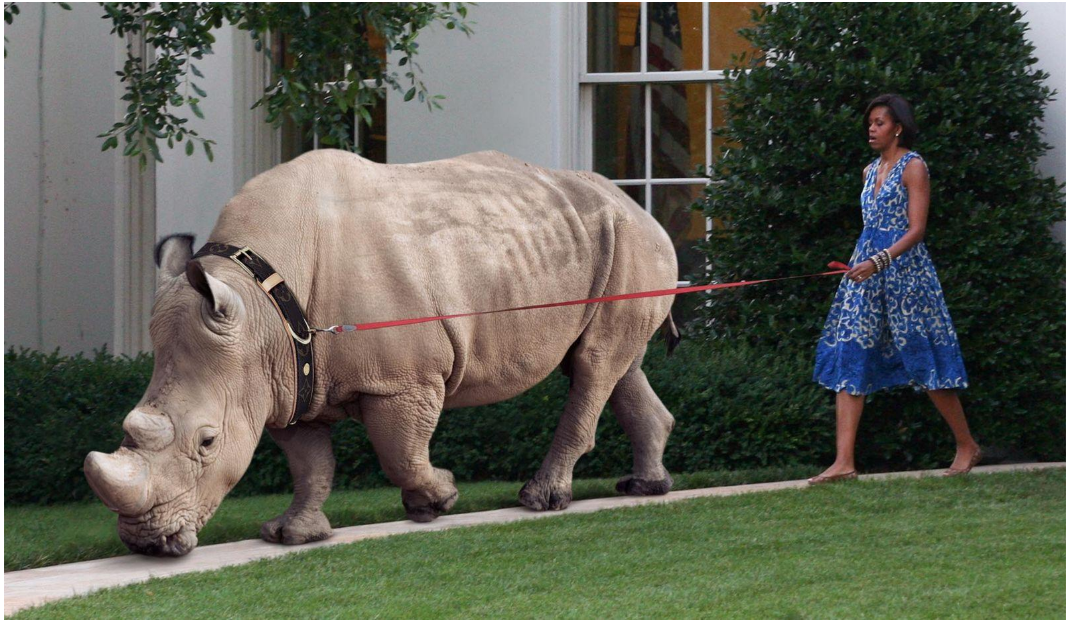
- COLUMB:
  - The splicing forgery regions are simple, large, and meaningless
  - About 180 samples
- CASIA:
  - The splicing forgery regions are objects, which are small and elaborated
  - About 850 samples

New datasets used:

- InWild: (Requires data augmentation)
  - High resolution and elaborated images
  - 200 samples
- New CASIA: (No data augmentation required)
  - Over 7000 samples

Datasets: **InWild - Data Augmentation**

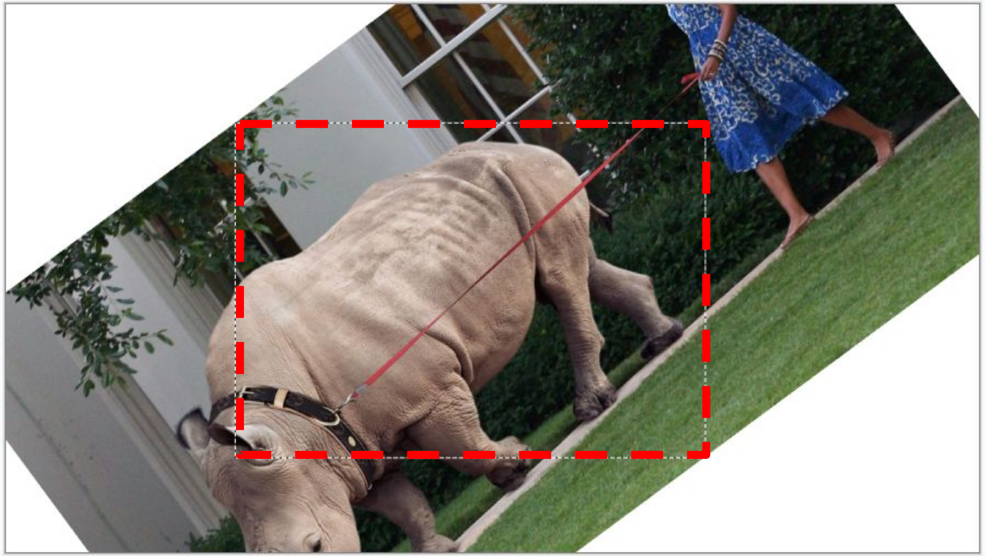
900

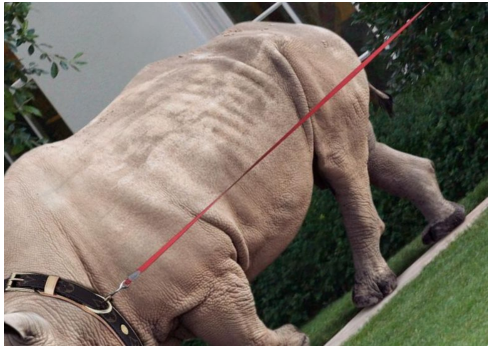


1600

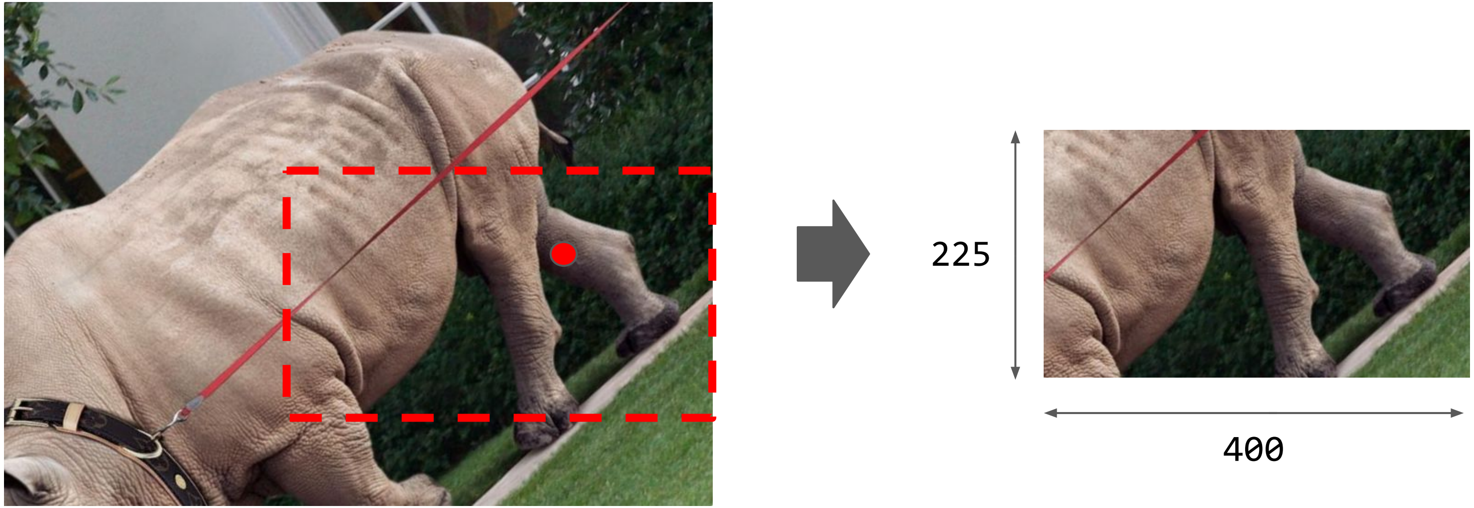
DATASET

Data Augmentation: **Step 1 - Rotate-Crop**





Data Augmentation: **Step 2 - Final Crop**



Data Augmentation: **Step 3 - imgaug**

ORIGINAL

GaussianBlur

Add (per_channel)

AdditiveGaussianNoise

JpegCompression

MotionBlur

Training - Parameters

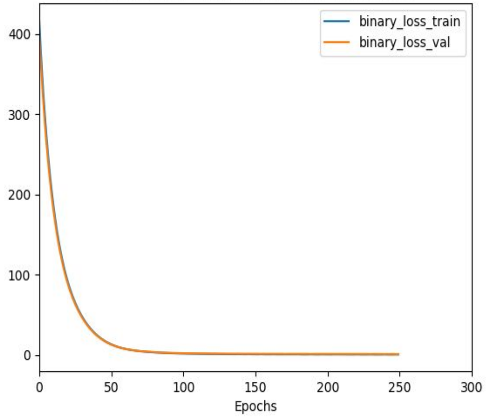
- Loss Function:
  - Binary Cross Entropy
- Normalization Layers:
  - Batch Normalization
  - Group Normalization
- Optimizers:
  - SGD
  - ADAM
  - RMSProp
- Weight initializer:
  - Glorot uniform initializer (Xavier)
  - PyTorch Conv2D initializer (uniform)
- Training in Google Colab
  - Nvidia Tesla K80
  - Nvidia Tesla T4
  - Nvidia Tesla P100

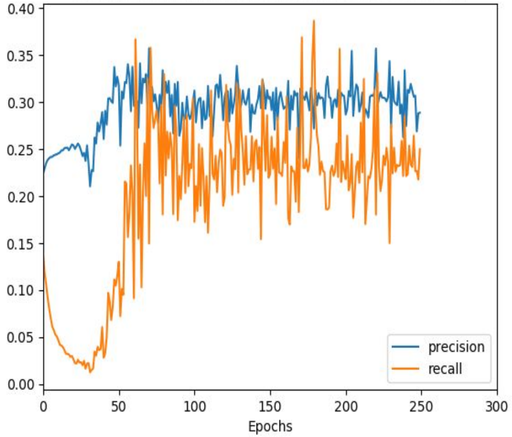
Training: **InWild Dataset**

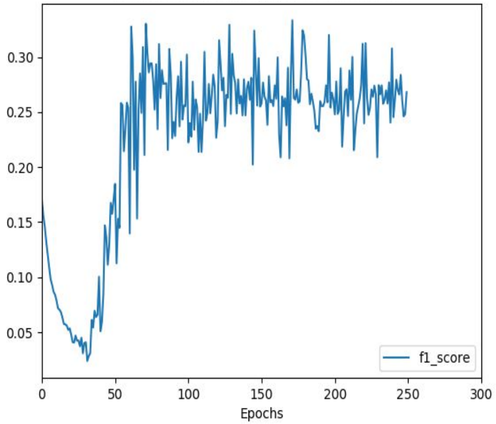
- Learning rates: {1E-01, 1E-03, 1E-05, 1E-06, 1E-07}
- L2 regularizations: {1E-01, 1E-02, 1E-03, 1E-04}

- Best result: LR=1E-05, L2=1E-01, ADAM, GN, BATCH_SIZE=10

  Test set: F1_score=0.23





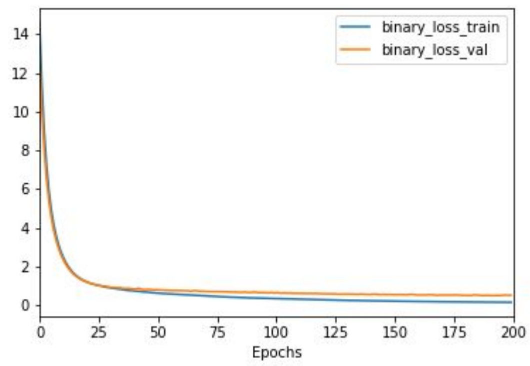


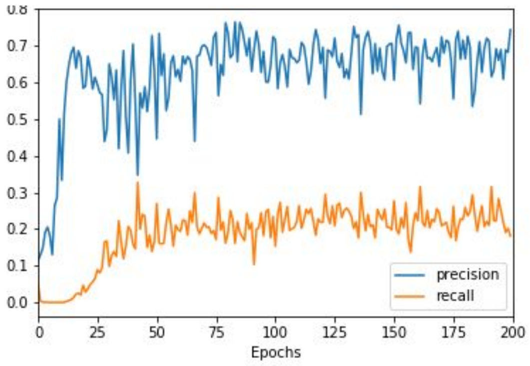
Training: **CASIA Dataset**

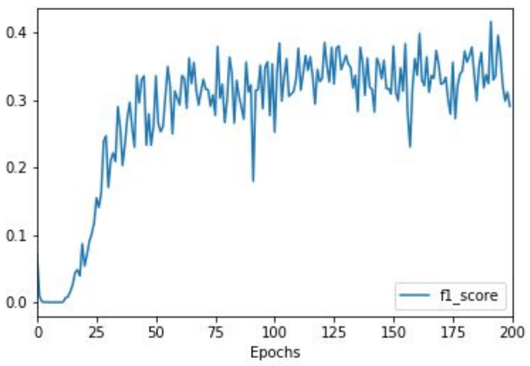
- Learning rates: {1E-04, 1E-05, 8E-06, 6E-06, 1E-06, 6E-07}
- L2 regularizations: {1E-01, 1E-02, 1E-03}

- Best result: LR=8E-06 L2=1E-02, RMSProp, PyTorch initializer, GN, BATCH_SIZE=10

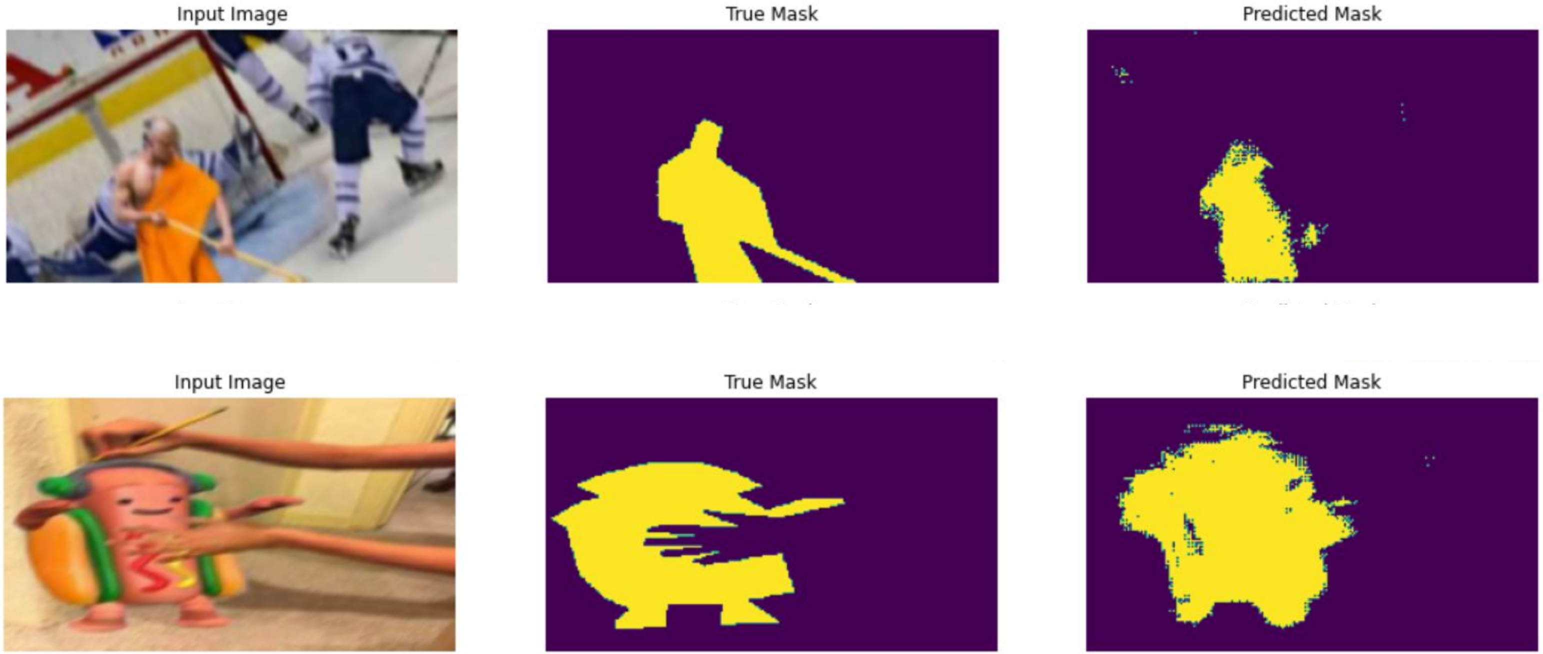
  Test set: F1_score=0.40



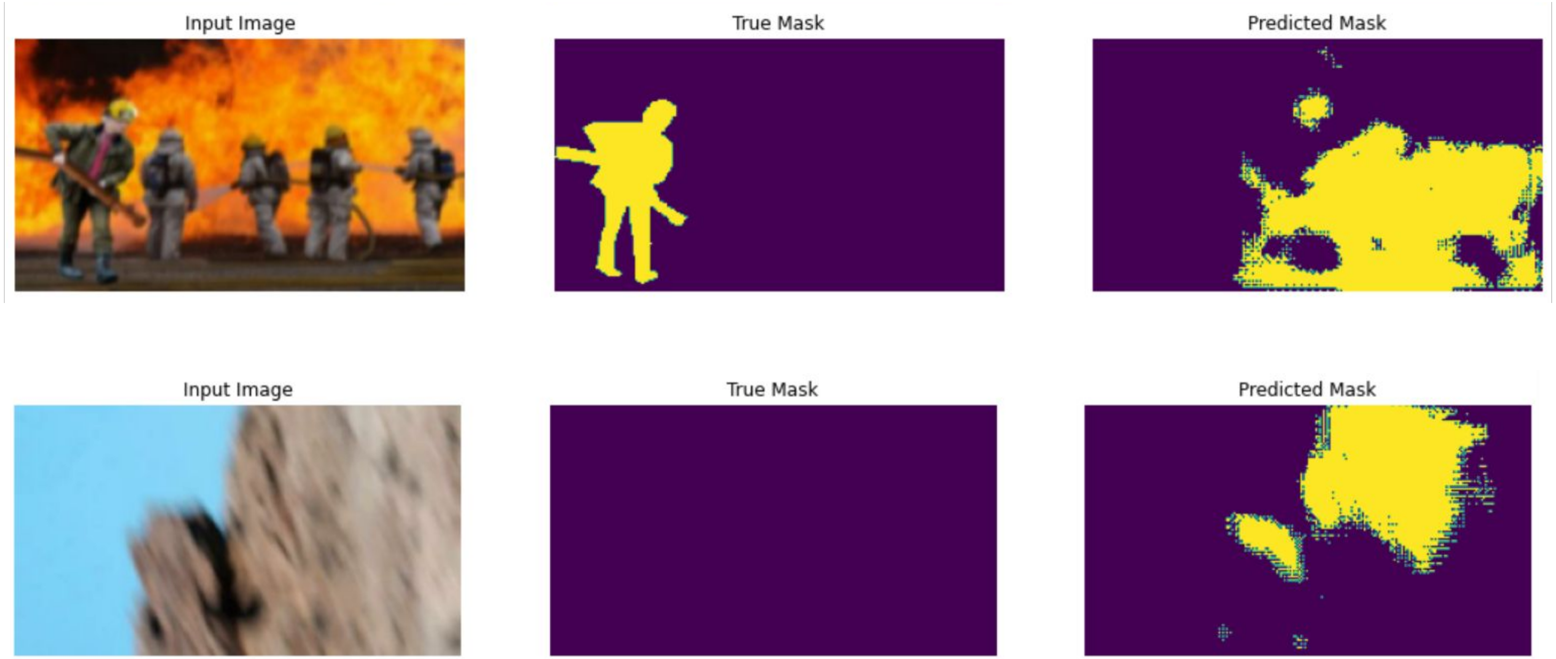




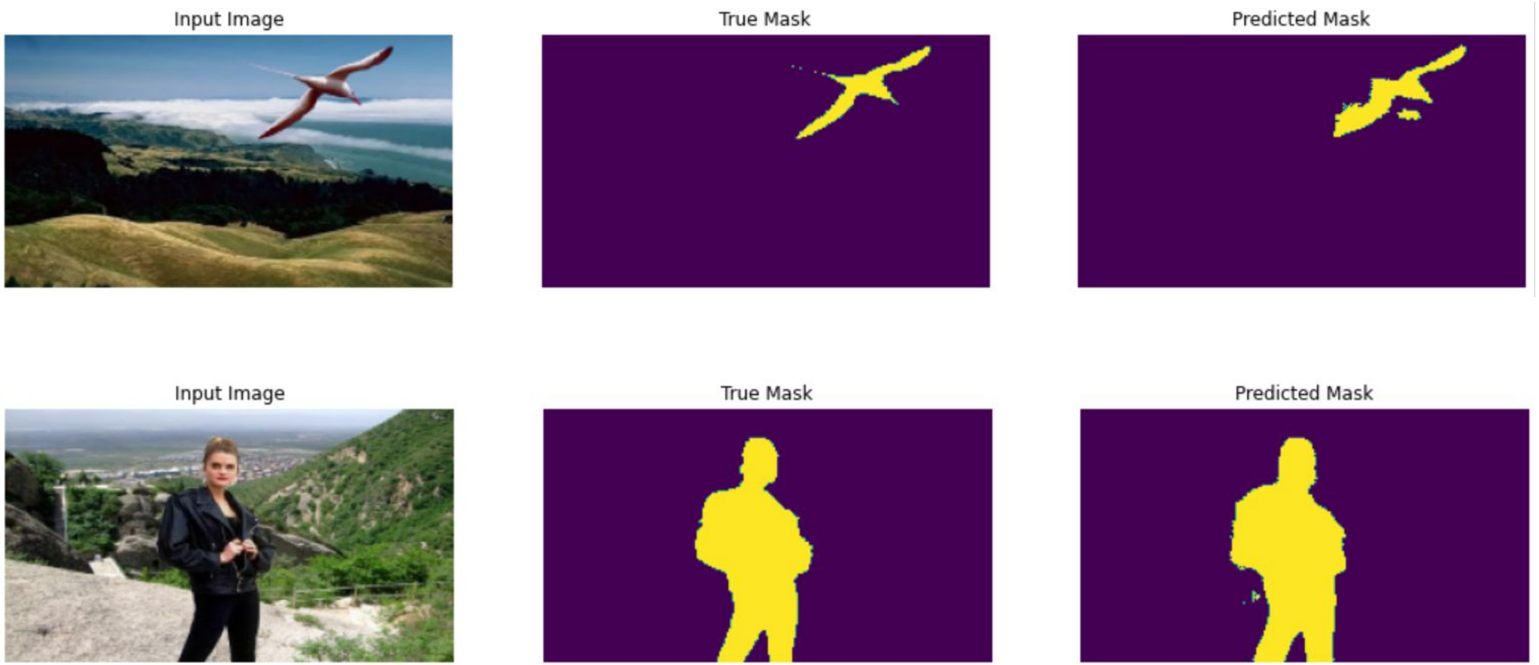
Results: **InWild - Good Predictions**



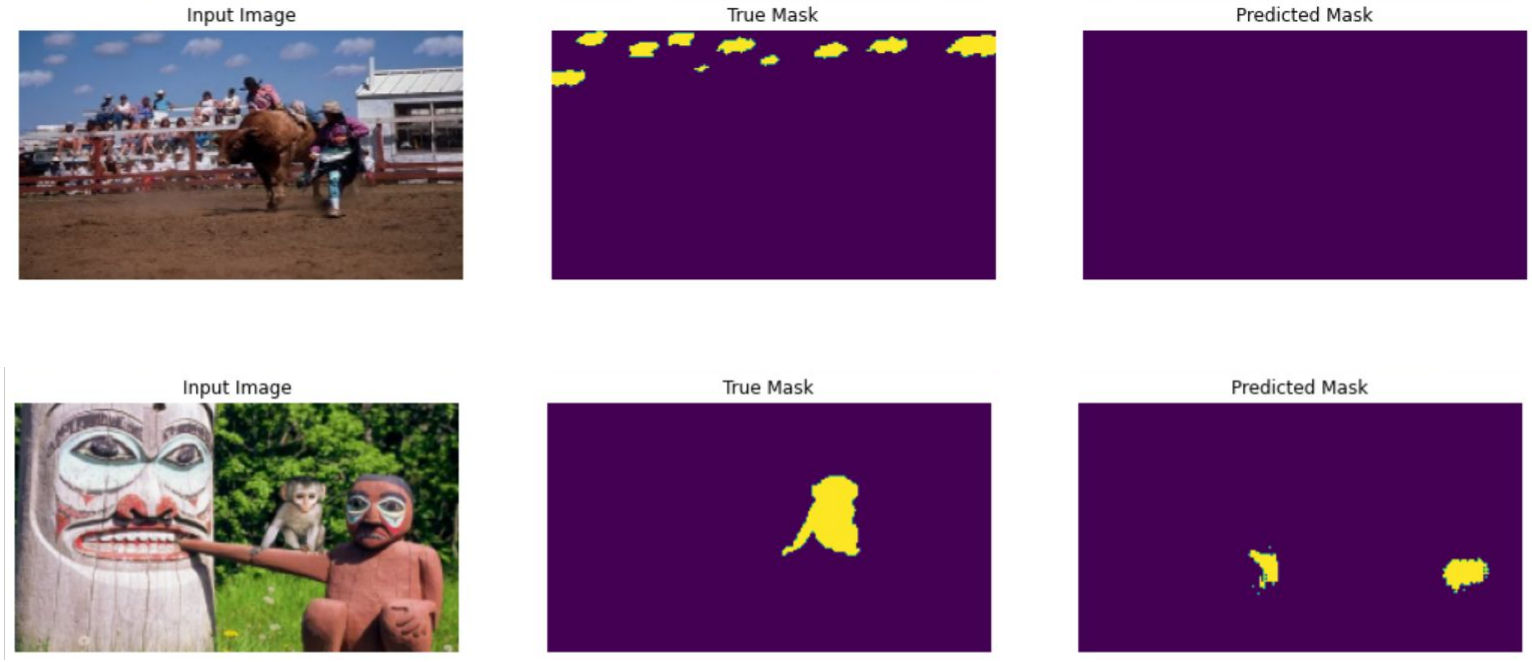
Results: **InWild - Wrong Predictions**



Results: **CASIA - Good Predictions**



Results: **CASIA - Wrong Predictions**



Thanks for your attention