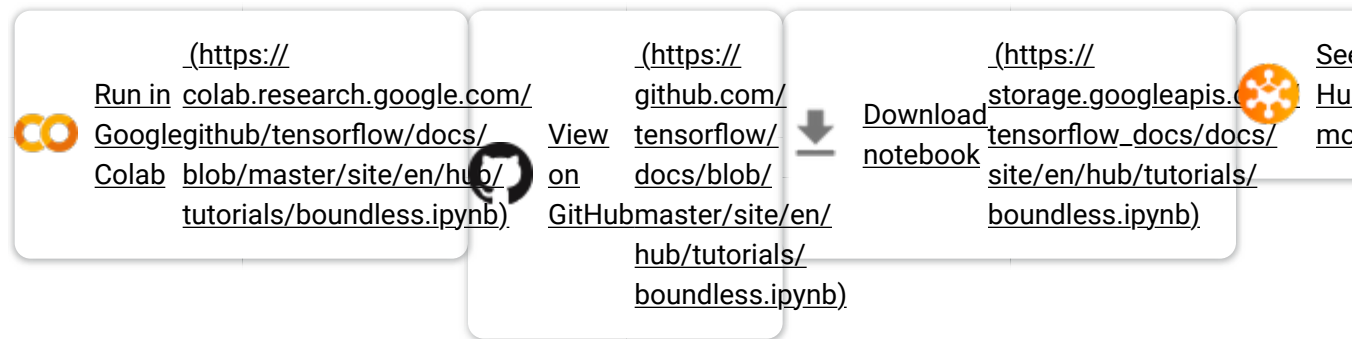


# Boundless Colab



Welcome to the Boundless model Colab! This notebook will take you through the steps of running the model on images and visualize the results.

## Overview

Boundless is a model for image extrapolation. This model takes an image, internally masks a portion of it ( $1/2$  (<https://tfhub.dev/google/boundless/half/1>),  $1/4$  (<https://tfhub.dev/google/boundless/quarter/1>),  $3/4$  ([https://tfhub.dev/google/boundless/three\\_quarter/1](https://tfhub.dev/google/boundless/three_quarter/1))) and completes the masked part. For more details refer to [Boundless: Generative Adversarial Networks for Image Extension](https://arxiv.org/pdf/1908.07007.pdf) (<https://arxiv.org/pdf/1908.07007.pdf>) or the model documentation on TensorFlow Hub.

## Imports and setup

Start with the base imports:

```
import tensorflow as tf
import tensorflow_hub as hub
from io import BytesIO
from PIL import Image as PilImage
import numpy as np
from matplotlib import pyplot as plt
from six.moves.urllib.request import urlopen
```

## Create a function for reading an image

Create a utility function to help load an image and format it for the model (257x257x3). This method will also crop the image to a square to avoid distortion and you can use it with local images or from the internet.

```
def read_image(filename):
    fd = None
    if(filename.startswith('http')):
        fd = urlopen(filename)
    else:
        fd = tf.io.gfile.GFile(filename, 'rb')

    pil_image = PilImage.open(fd)
    width, height = pil_image.size
    # crop to make the image square
    pil_image = pil_image.crop((0, 0, height, height))
    pil_image = pil_image.resize((257,257),PilImage.LANCZOS)
    image_unscaled = np.array(pil_image)
    image_np = np.expand_dims(
        image_unscaled.astype(np.float32) / 255., axis=0)
    return image_np
```

## Create a visualization function

Create a visualization function to show the original image side-by-side with the masked version and the "filled" version, both generated by the model.

```
def visualize_output_comparison(img_original, img_masked, img_filled):
    plt.figure(figsize=(24,12))
    plt.subplot(131)
    plt.imshow((np.squeeze(img_original)))
    plt.title("Original", fontsize=24)
    plt.axis('off')
    plt.subplot(132)
    plt.imshow((np.squeeze(img_masked)))
    plt.title("Masked", fontsize=24)
```

```
plt.axis('off')
plt.subplot(133)
plt.imshow((np.squeeze(img_filled)))
plt.title("Generated", fontsize=24)
plt.axis('off')
plt.show()
```

## Load an image

Now you can load a sample image. Feel free to use your own image by uploading it to the Colab notebook. Remember that the model may have some limitations regarding human images.

```
wikimedia = "https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/Nusfjord_
# wikimedia = "https://upload.wikimedia.org/wikipedia/commons/thumb/4/47/Beech_f
# wikimedia = "https://upload.wikimedia.org/wikipedia/commons/thumb/b/b2/Marmola
# wikimedia = "https://upload.wikimedia.org/wikipedia/commons/thumb/9/9d/Aegina_

input_img = read_image(wikimedia)
```

## Select a model from TensorFlow Hub

On TensorFlow Hub there are three versions of the Boundless model: Half, Quarter and Three Quarters. In the following cell you can choose any of the models and apply them on your image. If you want to pick another model, select it below and then run the following cells.

### Model Selection

Toggle code

```
model_name = 'Boundless Quarter' # @param ['Boundless Half', 'Boundless Quarter'
model_handle_map = {
    'Boundless Half' : 'https://tfhub.dev/google/boundless/half/1',
```

```
'Boundless Quarter' : 'https://tfhub.dev/google/boundless/quarter/1',
'Boundless Three Quarters' : 'https://tfhub.dev/google/boundless/three_quart
}

model_handle = model_handle_map[model_name]
```

After choosing your model, you can load it from TensorFlow Hub.

**Note:** You can point to a model handle to read the model's documentation.

```
print("Loading model {} ({}).format(model_name, model_handle))
model = hub.load(model_handle)
```

```
Loading model Boundless Quarter (https://tfhub.dev/google/boundless/quarter/1)
2024-03-09 13:56:07.413611: E external/local_xla/xla/stream_executor/cuda/cuda_d
```

## Perform inference

The boundless model has two outputs:

- The input image with a mask applied
- The masked image with the extrapolation to complete it

You can compare these two images with a visualization as follows:

```
result = model.signatures['default'](tf.constant(input_img))
generated_image = result['default']
masked_image = result['masked_image']

visualize_output_comparison(input_img, masked_image, generated_image)
```

Original	Masked	Generated
		



Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2024-03-09 UTC.