



Deep Learning Toolkit *(Gradio & Hugging Face)*

Rowel Atienza, PhD

University of the Philippines

github.com/roatienza

2022

Outline

Environment, Code Editor

Python

Tensor libraries – numpy, einsum, einops

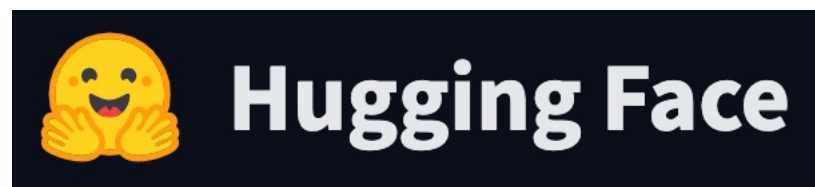
PyTorch, Timm

Hugging Face (HF), Gradio, Streamlit

HF Accelerator, GitHub

Machines – Colab, DeepNote, Kaggle, SageMaker

Other tools

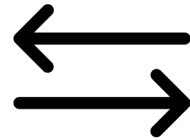
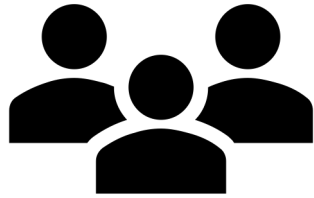


Gradio & Hugging Face

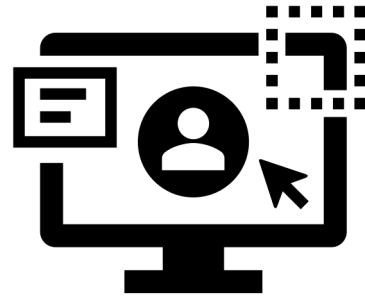
<https://gradio.app/>

<https://huggingface.co/>

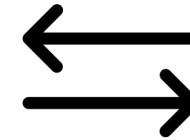
Users



Front-End



Back-End



Why Gradio?

Easy to use APIs to **build** model app UI

Easy to **demonstrate** model functionalities

Deploy model app through Hugging Face Spaces

Why Hugging Face?

A collection of pre-trained models in **Model Hub**

APIs to access and utilize pre-trained models using **Pipeline**

Hosting of deep learning apps using **Spaces**

Install

```
pip install transformers gradio
```

Introducing Gradio

```
import gradio as gr

def greet(name):
    return "Hello " + name + "!!"

gr.Interface(fn=greet,
             inputs="text",
             outputs="text").launch()
```


NAME

Clear

Submit

Flag

[view the api](#)  • built with [gradio](#) 

Using Gradio as ResNet18 Front End

1k Object Recognition

Demonstrates a pre-trained model from torchvision for image classification.

IMG

Drop Image Here
- or -
Click to Upload

Clear

Submit

Examples



Building Web App for ResNet18

```
gr.Interface(fn=classify,  
            inputs=gr.inputs.Image(shape=(224, 224)),  
            outputs=gr.outputs.Label(num_top_classes=5),  
            title="1k Object Recognition",  
            examples=[ 'wonder_cat.jpg', 'aki_dog.jpg', ],  
            description="Demonstrates a pre-trained model...",  
            allow_flagging="never"  
        ).launch(inbrowser=True)
```

```
def classify(img):  
    # By default, gradio image is numpy  
    img = torch.from_numpy(img)  
    # Numpy image is channel last. PyTorch is channel 1st.  
    img = rearrange(img, 'h w c -> c h w')  
  
    # The transforms before prediction  
    img = torchvision.transforms.Resize(256)(img)  
    img = torchvision.transforms.CenterCrop(224)(img).float()/255.  
    img = normalize(img)  
  
    # We insert batch size of 1  
    img = rearrange(img, 'c h w -> 1 c h w')
```

```
# The actual prediction
with torch.no_grad():
    pred = resnet(img)

# Convert the prediction to probabilities
pred = torch.nn.functional.softmax(pred, dim=1)
# Remove the batch dim. torch.squeeze() can also be used.
pred = rearrange(pred, "1 j->j")

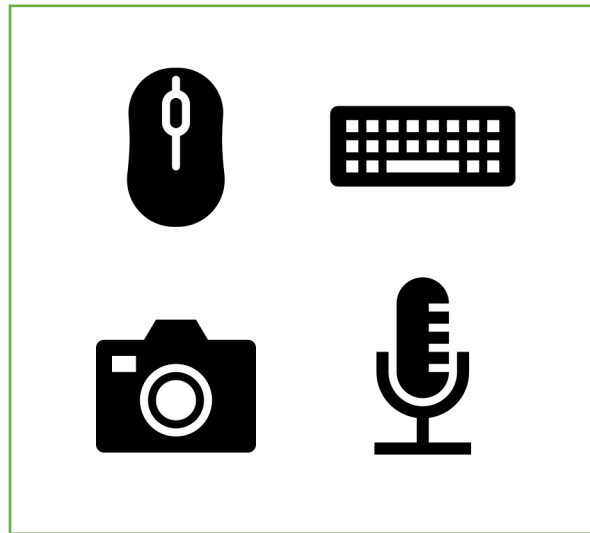
# torch to numpy space
pred = pred.cpu().numpy()

return {labels[i]: float(pred[i]) for i in range(1000)}
```

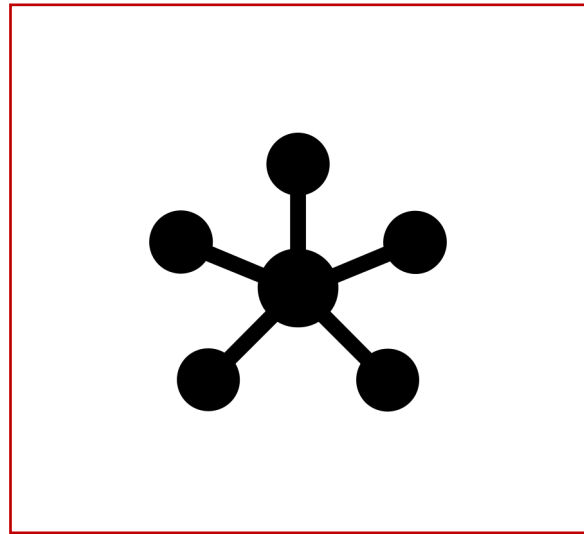
Pipeline

https://huggingface.co/docs/transformers/main_classes/pipelines

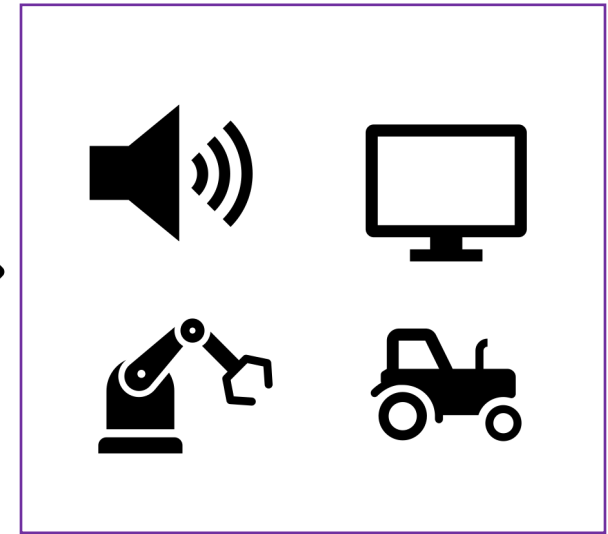
Pipeline



Input Pre-Processing



Model Inference



Output Post-Processing

Pipeline + Gradio

```
import gradio as gr
from transformers import pipeline

pipe = pipeline(task="image-classification",
                model="microsoft/beit-base-patch16-224-pt22k-ft22k")

gr.Interface.from_pipeline(pipe,
    title="22k Image Classification",
    description="Object Recognition using Microsoft BEIT",
    examples = ['wonder_cat.jpg', 'aki_dog.jpg', ],
    allow_flagging="never").launch(inbrowser=True)
```

22k Image Classification

Object Recognition using Microsoft BEiT

INPUT IMAGE

Drop Image Here
- or -
Click to Upload

Clear

Submit

Examples



Model Hub: <https://huggingface.co/models>

Natural Language Processing



Fill-Mask



Question Answering



Summarization



Table Question Answering



Text Classification



Text Generation



Text2Text Generation



Token Classification



Translation



Zero-Shot Classification



Sentence Similarity



Conversational



Feature Extraction

Audio



Text-to-Speech



Automatic Speech Recognition



Audio-to-Audio



Audio Classification



Voice Activity Detection

Computer Vision



Image Classification



Object Detection



Image Segmentation



Text-to-Image



Image-to-Text

Spaces

<https://huggingface.co/spaces>

<https://huggingface.co/spaces/rowel/22k-image-classification>

🌈 Spaces: 🧑 rowel/22k-image-classification 📄

🤍 like 1

📖 See logs

● Running

📦 App

📁 Files and versions

⚙️ Settings

📦 Linked Models

22k Image Classification

Object Recognition using Microsoft BEiT

Input Image

Drop Image Here
- or -
Click to Upload

Clear

Submit

Examples





Models

Datasets

Spaces

Docs



Solutions



Pricing



⌵







Tasks


 Fill-Mask  Question Answering

 Summarization  Table Question Answering



 Text Classification  Text Generation

 Text2Text Generation  Token Classification




 Translation  Zero-Shot Classification





 Sentence Similarity + 13


Libraries

 PyTorch  TensorFlow  JAX + 24

Datasets

 wikipedia  common_voice  squad

 bookcorpus  c4  glue  conll2003

 dcep europarl jrc-acquis + 810








Languages

 en  es  fr  de  zh  sv  fi  ja + 169

Licenses

Models 31,626

↑↓ Sort: Most Downloads

distilgpt2 Text Generation • Updated May 21, 2021 • ↓ 24.5M • ♥ 23**bert-base-uncased** Fill-Mask • Updated May 19, 2021 • ↓ 14.4M • ♥ 111 **cross-encoder/ms-marco-MiniLM-L-12-v2** Text Classification • Updated Aug 5, 2021 • ↓ 13.2M • ♥ 3**gpt2** Text Generation • Updated May 20, 2021 • ↓ 12.2M • ♥ 60 **Helsinki-NLP/opus-mt-zh-en** Translation • Updated Feb 27, 2021 • ↓ 7.09M • ♥ 16**xlm-roberta-large-finetuned-conll103-english** Token Classification • Updated Oct 12, 2020 • ↓ 5.67M • ♥ 11**distilbert-base-uncased** Fill-Mask • Updated Aug 30, 2021 • ↓ 5.56M • ♥ 44



Search models, datasets, users...

Models

Datasets

Spaces

Docs

Solutions

Pricing



Spaces: rowel/demospace

private

No application file

App

Files and versions

Settings

Get started with your Space!

Your new space has been created, follow these steps to get started (or read our full [documentation](#))

Start by cloning this repo by using:

```
$ git clone https://huggingface.co/spaces/rowel/demospace
```

Create your Gradio app.py file:

```
import gradio as gr




def greet(name):
    return "Hello " + name + "!!"


iface = gr.Interface(fn=greet, inputs="text", outputs="text")
iface.launch()
```


Then commit and push:








 **Spaces:**  rowel/**demospace**  private  See logs  Running

 App  **Files and versions**  Settings

 main **demospace / requirements.txt**

 rowel Update requirements.txt  12aea94 less than a minute ago

 raw  history  blame  edit  delete 18 Bytes

```
1 torch
2 transformers
```


App

Files and versions

Settings

Repo secrets

No secrets

rowel@eee.upd.edu.ph

.....

Add

Restart this Space

Click this button to trigger a reboot of your Space

Restart this Space

Change space visibility

This space is currently **private**. Only you (personal space) or members of your organization (organization space) can see and commit to this space.

Make this space public

Automatic Speech Recognition (ASR)

<https://huggingface.co/spaces/rowel/asr>

ASR

```
import gradio as gr
from transformers import pipeline

pipe = pipeline(task="automatic-speech-recognition",
                model="facebook/s2t-medium-librispeech-asr")

gr.Interface.from_pipeline(pipe,
                           title="Automatic Speech Recognition (ASR)",
                           description="Using pipeline with Facebook S2T for ASR.",
                           examples=[ 'data/ljspeech.wav' ],
                           ).launch(inbrowser=True)
```

Automatic Speech Recognition (ASR)

Using pipeline with Facebook S2T for ASR.

Input

Record

Clear

Submit

Examples

data/ljspeech.wav

Author: Rowel Atienza

Text to Speech (TTS)

<https://huggingface.co/spaces/rowel/tts>

TTS

```
import gradio as gr
```

```
gr.Interface.load(  
    "huggingface/facebook/fastspeech2-en-ljspeech",  
    description="TTS using FastSpeech2",  
    title="Text to Speech (TTS)",  
    examples=[["The quick brown fox jumps over the lazy dog."]]  
) .launch()
```

Text to Speech (TTS)

TTS using FastSpeech2

Input



Clear

Submit

Examples

The quick brown fox jumps over the lazy dog.

Text Generation

Text Generation using GPT2

```
import gradio as gr
```

```
gr.Interface.load("huggingface/gpt2",  
                  title="Text Generation",  
                  description="Using GPT2.",  
                  allow_flagging="never"  
                ).launch(inbrowser=True)
```

Text Generation

Using GPT2.

INPUT

Clear

Submit

[view the api](#)  • built with [gradio](#) 

Gradio Models in Series

Text Generation to Speech

```
textgen = gr.Interface.load("huggingface/EleutherAI/gpt-neo-2.7B")

tts = gr.Interface.load("huggingface/facebook/fastspeech2-en-ljspeech")

iface = gr.Series(textgen, tts)

iface.title = "Generated Text to Speech"
iface.allow_flagging = "never"
iface.launch(inbrowser=True)
```

Generated Text to Speech

INPUT

The |

Clear

Submit

[view the api](#) 🔧 • built with [gradio](#) 📦

End

https://github.com/roatienza/Deep-Learning-Experiments/blob/master/versions/2022/tools/python/gradio_demo.ipynb