

```
!pip install datasets diffusers transformers accelerate

Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from datasets) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from datasets) (6.0.3)
Requirement already satisfied: importlib_metadata in /usr/local/lib/python3.12/dist-packages (from diffusers) (8.7.1)
Requirement already satisfied: httpx<1.0.0 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.28.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from diffusers) (2025.11.3)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.7.0)
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (from diffusers) (11.3.0)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers)
Requirement already satisfied: typer-slim in /usr/local/lib/python3.12/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from accelerate) (2.9.0+cu126)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=2
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (4.12.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (2026.1.4)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (1.0
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (3.11)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1.0.0->diff
Requirement already satisfied: hf-xet<2.0.0,>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub)>=0.24
Requirement already satisfied: shellingham in /usr/local/lib/python3.12/dist-packages (from huggingface-hub)>=0.24.0->data
Requirement already satisfied: typing-extensions>=4.1.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub)>=
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests)>=2.32.2-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests)>=2.32.2->data
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->accelerate) (75.2
Requirement already satisfied: sympy=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->accelerate) (1
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->accelerate)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->accelerate) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch)>=2
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0-
Requirement already satisfied: nvidia-cUBLAS-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0-
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.
Requirement already satisfied: nvidia-cusparseLTLT-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->acc
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->ac
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0-
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch)>=2.0.0->accelerate) (3
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib_metadata->diffusers)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.3)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim->transformers) (8.
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a
Requirement already satisfied: yarl<2.0,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil)>=2.8.2->pandas->d
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=2
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch>=2.0.0->acce
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np
import matplotlib.pyplot as plt
from datasets import load_dataset
# 1. Load Dataset
print("Memuat dataset...")
raw_dataset = load_dataset("reach-vb/pokemon-blip-captions", split="train")
# 2. Ambil daftar caption untuk proses adaptasi teks
all_captions = [item['text'] for item in raw_dataset]
# 3. Setup Text Vectorization
max_tokens = 5000
seq_len = 20
text_vectorizer = layers.TextVectorization(
    max_tokens=max_tokens,
    output_sequence_length=seq_len,
)
# Proses Adapt (Mempelajari kosakata dari dataset)
text_vectorizer.adapt(all_captions)
vocab = text_vectorizer.get_vocabulary()
print(f"Kamus Teks Berhasil Dibuat. Jumlah kosakata: {len(vocab)}")
print("Contoh 10 kata pertama:", vocab[:10])
def preprocess_fn(item):
    # Proses caption
```

```

# Proses Gambar
image = item['image'].convert("RGB").resize((64, 64))
image = np.array(image) / 255.0 # Normalisasi0-1
# Proses Teks
caption = item['text']
return caption, image

# Membuat generator dataset
def gen():
    for item in raw_dataset:
        yield preprocess_fn(item)

# Membuat tf.data.Dataset
train_ds= tf.data.Dataset.from_generator(
gen,
output_signature=(
tf.TensorSpec(shape=(), dtype=tf.string),
tf.TensorSpec(shape=(64, 64, 3), dtype=tf.float32)
)
)

# Batching dan Transformasi Teks ke Angka
train_ds= train_ds.map(lambda x, y: (text_vectorizer(x), y))
train_ds = train_ds.batch(16).shuffle(100).prefetch(tf.data.AUTOTUNE)

class PokemonTrainer(keras.Model):
    def __init__(self, transformer, vqvae_encoder):
        super().__init__()
        self.transformer= transformer
        self.vqvae_encoder= vqvae_encoder
        self.loss_tracker= keras.metrics.Mean(name="loss")
    def train_step(self, data):
        text_tokens, images = data
        # 1. Ubah gambar asli menjadi token visual menggunakan encoder
        # Kita simulasi kandungan output dummy sesuai ukuran latent grid (misal 16x16)
        visual_tokens= tf.random.uniform((tf.shape(images)[0], 256), minval=0, maxval=1024, dtype=tf.int32)
        # 2. Siapkan input dan target (Autoregressive)
        vis_input= visual_tokens[:, :-1]
        vis_target= visual_tokens[:, 1:]
        with tf.GradientTape() as tape:
            # Prediksi
            preds = self.transformer([text_tokens, vis_input], training=True)
            # Hitung Loss
            loss = keras.losses.sparse_categorical_crossentropy(vis_target, preds, from_logits=True)
        grads = tape.gradient(loss, self.transformer.trainable_variables)
        self.optimizer.apply_gradients(zip(grads, self.transformer.trainable_variables))
        self.loss_tracker.update_state(loss)
        return {"loss": self.loss_tracker.result()}

# --- Placeholder Model Definitions (to resolve NameError) ---
# Define a placeholder Transformer Model
# This is a simplified model to pass the NameError, and will need a full implementation.
num_visual_tokens_in_vocab = 1024 # Based on tf.random.uniform maxval
def build_transformer_model(max_tokens, seq_len, num_visual_tokens):
    text_input = keras.Input(shape=(seq_len,), dtype=tf.int32, name="text_input")
    visual_input = keras.Input(shape=(255,), dtype=tf.int32, name="visual_sequence_input") # 256-1

    # Dummy embeddings and dense layer to match expected output shape for loss
    x = layers.Embedding(max_tokens, 64)(text_input)
    y = layers.Embedding(num_visual_tokens, 64)(visual_input)

    # A very simple way to combine and produce an output of required shape
    # In a real transformer, this would involve complex attention mechanisms.
    combined_features = layers.GlobalAveragePooling1D()(y) # (batch_size, 64)
    output_logits = layers.Dense(255 * num_visual_tokens)(combined_features) # (batch_size, 255 * 1024)
    output = layers.Reshape((255, num_visual_tokens))(output_logits) # (batch_size, 255, 1024)

    return keras.Model(inputs=[text_input, visual_input], outputs=output)

# Placeholder VQVAE Encoder Model
# This is a dummy encoder, as visual_tokens are randomly generated in train_step.
# It will need a full implementation.
def build_vqvae_encoder():
    input_image = keras.Input(shape=(64, 64, 3), name="input_image")
    x = layers.Conv2D(32, 3, activation="relu", strides=2, padding="same")(input_image)
    x = layers.Conv2D(64, 3, activation="relu", strides=2, padding="same")(x)
    x = layers.Flatten()(x)
    output = layers.Dense(256, activation="relu")(x) # Dummy output for latent vector
    return keras.Model(inputs=input_image, outputs=output)

transformer_model = build_transformer_model(max_tokens, seq_len, num_visual_tokens_in_vocab)
vqvae_encoder = build_vqvae_encoder()

# Placeholder for generate_image_tokens function
def generate_image_tokens(transformer_model, tokenized_text, num_tokens_to_generate, num_visual_tokens_in_vocab):
    # This is a dummy function for demonstration. A real implementation would involve autoregressive generation.
    return tf.random.uniform(shape=(tf.shape(tokenized_text)[0], num_tokens_to_generate), minval=0, maxval=num_visual_tokens_in_vocab)

```

```
# Placeholder for decode_to_real_image function
def decode_to_real_image(gen_vis_tokens):
    # This is a dummy function. A real implementation would use a VAE decoder to reconstruct an image.
    # Returns a random image for now.
    return np.random.rand(64, 64, 3)
# -----

# Inisialisasidan Compile
trainer = PokemonTrainer(transformer_model, vqvae_encoder)
trainer.compile(optimizer=keras.optimizers.Adam(learning_rate=1e-4))
# 5. JalankanTraining
print("MemulaiPelatihan...")
trainer.fit(train_ds, epochs=10)
def generate_pokemon(prompt):
    # 1. Ubahtekskeangka
    tokenized_text= text_vectorizer([prompt])
    # 2. Generate token visual (Autoregressive)
    # Gunakanfungsigenerate_image_tokensyang kitabuatsebelumnya
    gen_vis_tokens= generate_image_tokens(transformer_model, tokenized_text, 256, 1024)
    # 3. Decode jadiGambar menggunakanPre-trained VAE
    # Gunakanfungsidecode_to_real_imageyang memanggilAutoencoderKL
    final_image= decode_to_real_image(gen_vis_tokens)
    plt.imshow(final_image)
    plt.title(prompt)
    plt.axis("off")
    plt.show()
# TEST
generate_pokemon("a pink cute pokemon")
```

Memuatdataset...

Kamus Teks Berhasil Dibuat. Jumlah kosakata: 359
 Contoh 10 kata pertama: ['[', '[UNK]', np.str_('a'), np.str_('with'), np.str_('of'), np.str_('cartoon'), np.str_('and'), np.str_(''), np.str_('')]
 MemulaiPelatihan...
 Epoch 1/10
 53/53 17s 19ms/step - loss: 6.9315
 Epoch 2/10
 53/53 13s 7ms/step - loss: 6.9315
 Epoch 3/10
 53/53 14s 7ms/step - loss: 6.9315
 Epoch 4/10
 53/53 14s 6ms/step - loss: 6.9315
 Epoch 5/10
 53/53 14s 7ms/step - loss: 6.9315
 Epoch 6/10
 53/53 14s 7ms/step - loss: 6.9315
 Epoch 7/10
 53/53 16s 6ms/step - loss: 6.9315
 Epoch 8/10
 53/53 14s 7ms/step - loss: 6.9315
 Epoch 9/10
 53/53 14s 8ms/step - loss: 6.9315
 Epoch 10/10
 53/53 14s 6ms/step - loss: 6.9315

a pink cute pokemon



