

```
# Install necessary libraries
!pip install tensorflow keras matplotlib scikit-image tensorflow-addons gdown

# Create folders for structure
!mkdir -p SRGAN_Project/models SRGAN_Project/utils SRGAN_Project/results DIV2K

→ Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/dist-packages (2.18.0)
Requirement already satisfied: keras in /usr/local/lib/python3.11/dist-packages (3.8.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.11/dist-packages (0.25.2)
Collecting tensorflow-addons
  Downloading tensorflow_addons-0.23.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.8 kB)
Requirement already satisfied: gdown in /usr/local/lib/python3.11/dist-packages (5.2.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.2)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<6.0.0dev,>=3.20.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (4.13.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0)
Requirement already satisfied: numpy<2.1.0,>=1.26.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.0.2)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras) (0.0.9)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras) (0.15.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: scipy>=1.11.4 in /usr/local/lib/python3.11/dist-packages (from scikit-image) (1.14.1)
Requirement already satisfied: networkx>=3.0 in /usr/local/lib/python3.11/dist-packages (from scikit-image) (3.4.2)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.11/dist-packages (from scikit-image) (2.37.0)
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.11/dist-packages (from scikit-image) (2025.3.30)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.11/dist-packages (from scikit-image) (0.4)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow-addons)
  Downloading typeguard-2.13.3-py3-none-any.whl.metadata (3.6 kB)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from gdown) (4.13.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from gdown) (3.18.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from gdown) (4.67.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->tensorflow)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.4.0)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->gdown) (2.7)
```

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from glob import glob
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, PReLU, Add, UpSampling2D, LeakyReLU, Dense, Flatten, Concatenate
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.applications import VGG19
from tensorflow.keras.optimizers import Adam
from skimage.metrics import peak_signal_noise_ratio, structural_similarity

# Create folder for dataset
!mkdir -p DIV2K

# Download DIV2K high-resolution images (approx 7.5GB)
!wget -c https://data.vision.ee.ethz.ch/cvl/DIV2K/DIV2K_train_HR.zip -O DIV2K/DIV2K_train_HR.zip
```

```
# Unzip
!unzip -q DIV2K/DIV2K_train_HR.zip -d DIV2K/
```

→ --2025-04-25 10:51:59-- https://data.vision.ee.ethz.ch/cvl/DIV2K/DIV2K_train_HR.zip
 Resolving data.vision.ee.ethz.ch (data.vision.ee.ethz.ch)... 129.132.52.178, 2001:67c:10ec:36c2::178
 Connecting to data.vision.ee.ethz.ch (data.vision.ee.ethz.ch)|129.132.52.178|:443... connected.
 HTTP request sent, awaiting response... 200 OK
 Length: 3530603713 (3.3G) [application/zip]
 Saving to: 'DIV2K/DIV2K_train_HR.zip'

 DIV2K/DIV2K_train_HR 100%[=====] 3.29G 22.0MB/s in 2m 37s

 2025-04-25 10:54:37 (21.4 MB/s) - 'DIV2K/DIV2K_train_HR.zip' saved [3530603713/3530603713]

```
from glob import glob
import cv2
import numpy as np
from sklearn.model_selection import train_test_split

# Resize DIV2K HR images and generate LR
def preprocess_div2k(hr_folder, hr_size=(256, 256), lr_size=(64, 64), limit=800):
    hr_images, lr_images = [], []
    hr_paths = sorted(glob(f"{hr_folder}/DIV2K_train_HR/*.png"))[:limit]

    for path in hr_paths:
        img = cv2.imread(path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        hr = cv2.resize(img, hr_size)
        lr = cv2.resize(hr, lr_size, interpolation=cv2.INTER_CUBIC)
        hr_images.append(hr / 255.0)
        lr_images.append(lr / 255.0)

    return np.array(lr_images), np.array(hr_images)

# Load and split
x_lr, x_hr = preprocess_div2k("DIV2K")
x_hr_train, x_hr_val, x_lr_train, x_lr_val = train_test_split(x_hr, x_lr, test_size=0.1)
print(f"DIV2K Loaded → HR: {x_hr.shape}, LR: {x_lr.shape}")
```

→ DIV2K Loaded → HR: (800, 256, 256, 3), LR: (800, 64, 64, 3)

```
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, PReLU, Add, UpSampling2D
from tensorflow.keras.models import Model
```

```
def residual_block(x_input):
    x = Conv2D(64, kernel_size=3, strides=1, padding='same')(x_input)
    x = BatchNormalization()(x)
    x = PReLU(shared_axes=[1, 2])(x)
    x = Conv2D(64, kernel_size=3, strides=1, padding='same')(x)
    x = BatchNormalization()(x)
    x = Add()([x_input, x])
    return x

def build_generator():
    inputs = Input(shape=(64, 64, 3))
    x = Conv2D(64, 9, padding='same')(inputs)
    x = PReLU(shared_axes=[1, 2])(x)
    residual = x

    for _ in range(16):
        residual = residual_block(residual)

    x = Conv2D(64, 3, padding='same')(residual)
    x = BatchNormalization()(x)
    x = Add()([x, residual])

    for _ in range(2):
        x = UpSampling2D()(x)
        x = Conv2D(256, 3, padding='same')(x)
        x = PReLU(shared_axes=[1, 2])(x)

    outputs = Conv2D(3, 9, padding='same', activation='tanh')(x)
    return Model(inputs, outputs, name="SRGAN_Generator")

generator = build_generator()
generator.summary()
```

Model: "SRGAN_Generator"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 64, 64, 3)	0	-
conv2d (Conv2D)	(None, 64, 64, 64)	15,616	input_layer[0][0]
p_re_lu (PReLU)	(None, 64, 64, 64)	64	conv2d[0][0]
conv2d_1 (Conv2D)	(None, 64, 64, 64)	36,928	p_re_lu[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_1[0][0]
p_re_lu_1 (PReLU)	(None, 64, 64, 64)	64	batch_normalizat...
conv2d_2 (Conv2D)	(None, 64, 64, 64)	36,928	p_re_lu_1[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_2[0][0]
add (Add)	(None, 64, 64, 64)	0	p_re_lu[0][0], batch_normalizat...
conv2d_3 (Conv2D)	(None, 64, 64, 64)	36,928	add[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_3[0][0]
p_re_lu_2 (PReLU)	(None, 64, 64, 64)	64	batch_normalizat...
conv2d_4 (Conv2D)	(None, 64, 64, 64)	36,928	p_re_lu_2[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_4[0][0]
add_1 (Add)	(None, 64, 64, 64)	0	add[0][0], batch_normalizat...
conv2d_5 (Conv2D)	(None, 64, 64, 64)	36,928	add_1[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_5[0][0]
p_re_lu_3 (PReLU)	(None, 64, 64, 64)	64	batch_normalizat...
-----	-----	-----	-----

```
from tensorflow.keras.layers import LeakyReLU, Dense, Flatten
from tensorflow.keras.models import Sequential

def build_discriminator():
    model = Sequential()
    model.add(Conv2D(64, kernel_size=3, strides=1, padding='same', input_shape=(256, 256, 3)))
    model.add(LeakyReLU(alpha=0.2))

    filters = [64, 128, 128, 256, 256, 512, 512]
    strides = [2, 1, 2, 1, 2, 1, 2]

    for f, s in zip(filters, strides):
        model.add(Conv2D(f, 3, strides=s, padding='same'))
        model.add(BatchNormalization())
        model.add(LeakyReLU(alpha=0.2))

    model.add(Flatten())
    model.add(Dense(1024))
    model.add(LeakyReLU(alpha=0.2))
    model.add(Dense(1, activation='sigmoid'))

    return model

discriminator = build_discriminator()
discriminator.summary()
```

(BatchNormalizatio... 64)	-----	-----	-----
p_re_lu_5 (PReLU)	(None, 64, 64,	64	batch_normalizat...

<code>usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base.py:107: UserWarning: Do not pass an `input_shape`/`super().__init__(activity_regularizer=activity_regularizer, **kwargs)</code>	<code>/usr/local/lib/python3.11/dist-packages/keras/src/layers/activations/leaky_relu.py:41: UserWarning: Argument `alpha` is deprecated.</code>
<code>warnings.warn("Model: " + self.__class__.__name__)</code>	
<code>batch_normalization_0 (BatchNormalization) (None, 64, 64, 64) Output Shape Param #</code>	
<code>add_1 (Add) (Conv2D) (None, 64, 64, 64) (None, 256, 256, 64) add_3[0][0], 1,792</code>	<code>batch_normalization_1 (LeakyReLU) (None, 64, 64, 64) (None, 256, 256, 64) add_4[0][0] 0</code>
<code>conv2d_37 (Conv2D) (None, 64, 64, 64) (None, 128, 128, 64) 36,928</code>	<code>conv2d_11 (Conv2D) (None, 64, 64, 64) (None, 128, 128, 256) conv2d_11[0][0] 256</code>
<code>batch_normalization_33 (BatchNormalization) (None, 64, 64, 64) (None, 128, 128, 256) batch_normalization_33[0][0]</code>	<code>leaky_re_lu_6 (PReLU) (None, 64, 64, 64) (None, 128, 128, 64) batch_normalization_6[0][0]</code>
<code>conv2d_39 (Conv2D) (None, 64, 64, 64) (None, 128, 128, 128) p_re_lu_6[0][0] 73,856</code>	<code>conv2d_12 (Conv2D) (None, 64, 64, 64) (None, 128, 128, 128) p_re_lu_7[0][0] 512</code>
<code>batch_normalization_34 (BatchNormalization) (None, 64, 64, 64) (None, 128, 128, 128) batch_normalization_34[0][0]</code>	<code>batch_normalization_35 (BatchNormalization) (None, 64, 64, 64) (None, 64, 64, 128) conv2d_12[0][0] 0</code>
<code>leaky_re_lu_1 (LeakyReLU) (None, 64, 64, 64) (None, 128, 128, 128) 0</code>	<code>add_4 (Add) (Conv2D) (None, 64, 64, 64) (None, 64, 64, 128) add_4[0][0] 1,47,584</code>
<code>conv2d_40 (Conv2D) (None, 64, 64, 64) (None, 64, 64, 128) batch_normalization_40[0][0]</code>	<code>batch_normalization_36 (BatchNormalization) (None, 64, 64, 64) (None, 64, 64, 256) add_5[0][0] 512</code>
<code>batch_normalization_35 (BatchNormalization) (None, 64, 64, 64) (None, 64, 64, 128) conv2d_13[0][0] 0</code>	<code>leaky_re_lu_3 (LeakyReLU) (None, 64, 64, 64) (None, 64, 64, 256) conv2d_13[0][0] 0</code>
<code>conv2d_41 (Conv2D) (None, 64, 64, 64) (None, 64, 64, 256) 295,168</code>	<code>p_re_lu_7 (PReLU) (None, 64, 64, 64) (None, 64, 64, 256) batch_normalization_7[0][0] 1,024</code>
<code>batch_normalization_36 (BatchNormalization) (None, 64, 64, 64) (None, 64, 64, 256) p_re_lu_8[0][0] 0</code>	<code>conv2d_14 (Conv2D) (None, 64, 64, 64) (None, 32, 32, 256) conv2d_14[0][0] 590,080</code>
<code>batch_normalization_37 (BatchNormalization) (None, 64, 64, 64) (None, 32, 32, 256) 1,024</code>	<code>batch_normalization_37 (BatchNormalization) (None, 64, 64, 64) (None, 32, 32, 256) add_6[0][0], 0</code>
<code>add_6 (Add) (None, 64, 64, 64) (None, 32, 32, 256) batch_normalization_6[0][0]</code>	<code>leaky_re_lu_5 (LeakyReLU) (None, 64, 64, 64) (None, 32, 32, 256) batch_normalization_5[0][0]</code>
<code>conv2d_43 (Conv2D) (None, 64, 64, 64) (None, 32, 32, 512) add_6[0][0] 1,180,160</code>	<code>conv2d_15 (Conv2D) (None, 64, 64, 64) (None, 32, 32, 512) conv2d_15[0][0] 2,048</code>
<code>batch_normalization_38 (BatchNormalization) (None, 64, 64, 64) (None, 32, 32, 512) batch_normalization_8[0][0] 0</code>	<code>leaky_re_lu_6 (LeakyReLU) (None, 64, 64, 64) (None, 32, 32, 512) batch_normalization_6[0][0]</code>
<code>conv2d_44 (Conv2D) (None, 64, 64, 64) (None, 16, 16, 512) 2,359,808</code>	<code>p_re_lu_8 (PReLU) (None, 64, 64, 64) (None, 16, 16, 512) p_re_lu_8[0][0] 2,048</code>
<code>conv2d_16 (Conv2D) (None, 64, 64, 64) (None, 16, 16, 512) conv2d_16[0][0] 0</code>	<code>batch_normalization_39 (BatchNormalization) (None, 64, 64, 64) (None, 16, 16, 512) conv2d_16[0][0] 0</code>
<code>batch_normalization_40 (BatchNormalization) (None, 64, 64, 64) (None, 16, 16, 512) flatten (Flatten) (None, 64, 64, 131072) 0 add_6[0][0], 0</code>	<code>leaky_re_lu_7 (LeakyReLU) (None, 64, 64, 64) (None, 16, 16, 512) conv2d_17[0][0] 0</code>
<code>add_7 (Add) (None, 64, 64, 131072) (None, 1024) batch_normalization_7[0][0]</code>	<code>dense (Dense) (None, 64, 64, 1024) 134,218,752</code>
<code>conv2d_17 (Conv2D) (None, 64, 64, 64) (None, 1024) 36,928 add_7[0][0] 0</code>	<code>leaky_re_lu_8 (LeakyReLU) (None, 64, 64, 64) (None, 1024) 36,928 add_8[0][0] 0</code>
<code>dense_1 (Dense) (None, 64, 64, 1) 256 conv2d_17[0][0] 1,025</code>	<code>batch_normalization_41 (BatchNormalization) (None, 64, 64, 1) 256 conv2d_18[0][0] 0</code>

```
from tensorflow.keras.applications import VGG19
from tensorflow.keras.models import Model
```

```
def build_vgg():
    vgg = VGG19(weights="imagenet", include_top=False, input_shape=(256, 256, 3))
    vgg.trainable = False
    model = Model(inputs=vgg.input, outputs=vgg.get_layer("block5_conv4").output)
    return model
```

```
vgg = build_vgg()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_no_top_34684/80a34624tatio... (None, 64, 64, 0) 0s 0us/step 256 conv2d_19[0][0]
```

```
from tensorflow.keras.losses import BinaryCrossentropy, MeanSquaredError
from tensorflow.keras.optimizers import Adam
```

```

import numpy as np

g_optimizer = Adam(1e-4)
d_optimizer = Adam(1e-4)

bce = BinaryCrossentropy()
mse = MeanSquaredError()

@tf.function
@tf.function
def train_step(lr, hr):
    valid = tf.ones((lr.shape[0], 1)) * 0.9
    fake = tf.zeros((lr.shape[0], 1)) + 0.1
    valid += tf.random.uniform(valid.shape, 0.0, 0.05)
    fake += tf.random.uniform(fake.shape, 0.0, 0.05)

    with tf.GradientTape() as g_tape, tf.GradientTape() as d_tape:
        sr = generator(lr, training=True)
        real_output = discriminator(hr, training=True)
        fake_output = discriminator(sr, training=True)

        d_loss = bce(valid, real_output) + bce(fake, fake_output)

        sr_features = vgg(sr)
        hr_features = vgg(hr)
        content_loss = mse(hr_features, sr_features)
        g_loss = content_loss + 1e-3 * bce(valid, fake_output)

    grads_g = g_tape.gradient(g_loss, generator.trainable_variables)
    grads_d = d_tape.gradient(d_loss, discriminator.trainable_variables)
    g_optimizer.apply_gradients(zip(grads_g, generator.trainable_variables))
    d_optimizer.apply_gradients(zip(grads_d, discriminator.trainable_variables))

    return g_loss, d_loss

```

```

from skimage.metrics import peak_signal_noise_ratio as psnr_metric
from skimage.metrics import structural_similarity as ssim_metric
import matplotlib.pyplot as plt
import os

# Make sure result folder exists
os.makedirs("SRGAN_Project/results", exist_ok=True)

for epoch in range(30): # 30 epochs
    g_losses, d_losses = [], []

    for i in range(0, x_lr_train.shape[0], 8): # Batch size = 8
        lr_batch = x_lr_train[i:i+8]
        hr_batch = x_hr_train[i:i+8]
        g_loss, d_loss = train_step(lr_batch, hr_batch)
        g_losses.append(g_loss)
        d_losses.append(d_loss)

    # Predict and visualize after EVERY epoch
    sr_imgs = generator.predict(x_lr_val[:5])
    psnr_total, ssim_total = 0, 0

    fig, axs = plt.subplots(3, 5, figsize=(15, 7))
    for i in range(5):
        lr = x_lr_val[i]
        sr = np.clip(sr_imgs[i], 0, 1)
        hr = np.clip(x_hr_val[i], 0, 1)

        psnr_total += psnr_metric(hr, sr, data_range=1.0)
        ssim_total += ssim_metric(hr, sr, channel_axis=2, data_range=1.0)

        axs[0, i].imshow(lr)
        axs[0, i].set_title("Low Res")
        axs[0, i].axis('off')

        axs[1, i].imshow(sr)
        axs[1, i].set_title("SRGAN Output")
        axs[1, i].axis('off')

        axs[2, i].imshow(hr)
        axs[2, i].set_title("High Res (GT)")
        axs[2, i].axis('off')

    avg_psnr = psnr_total / 5
    avg_ssim = ssim_total / 5

```

```
plt.suptitle(f"Epoch {epoch+1} | PSNR: {avg_psnr:.4f} | SSIM: {avg_ssimm:.4f}", fontsize=16)
plt.tight_layout()

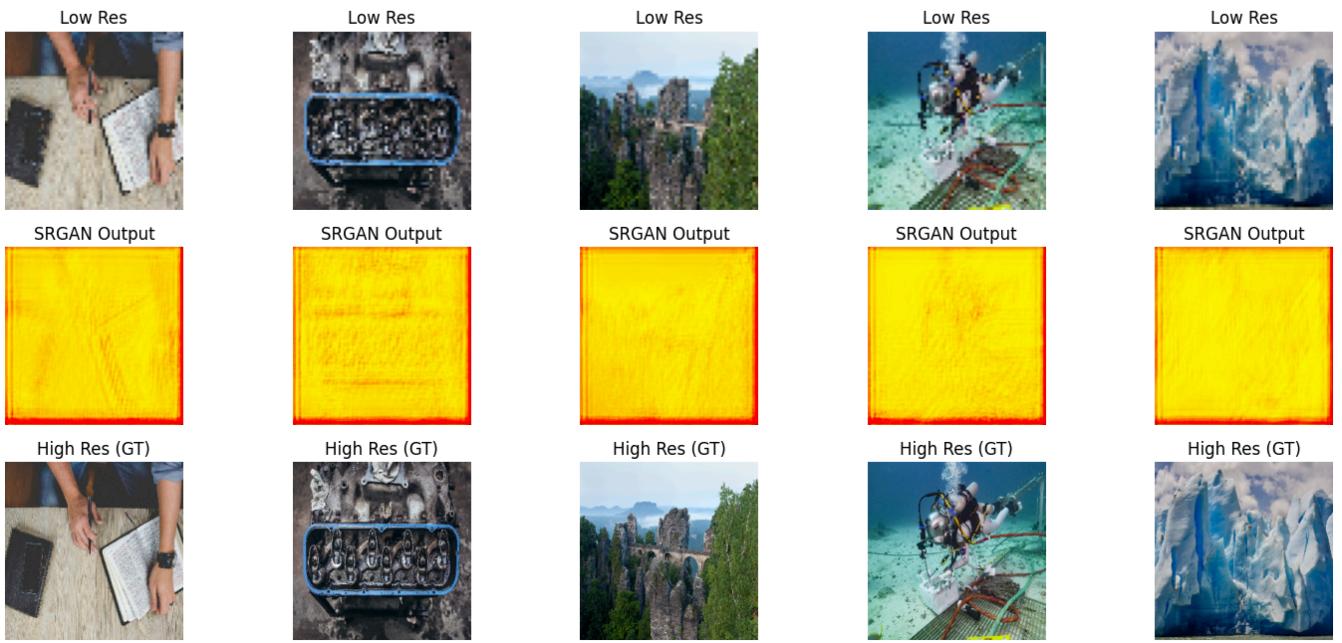
#  Save the image grid
plt.savefig(f"SRGAN_Project/results/grid_epoch_{epoch+1}.png")
plt.show()

#  Print loss
print(f"[Epoch {epoch+1:02d}] G Loss: {np.mean(g_losses):.4f} | D Loss: {np.mean(d_losses):.4f} | PSNR: {avg_psnr:.4f} | SSIM: {avg_ssimm:.4f}
```

	64)		
conv2d_32 (Conv2D)	(None, 64, 64, 64)	36,928	p_re_lu_16[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_32[0][0]
add_15 (Add)	(None, 64, 64, 64)	0	add_14[0][0], batch_normalizat...
conv2d_33 (Conv2D)	(None, 64, 64, 64)	36,928	add_15[0][0]
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256	conv2d_33[0][0]
add_16 (Add)	(None, 64, 64, 64)	0	batch_normalizat... add_15[0][0]
up_sampling2d (UpSampling2D)	(None, 128, 128, 64)	0	add_16[0][0]
conv2d_34 (Conv2D)	(None, 128, 128, 256)	147,712	up_sampling2d[0]...

1/1 9s 9s/step

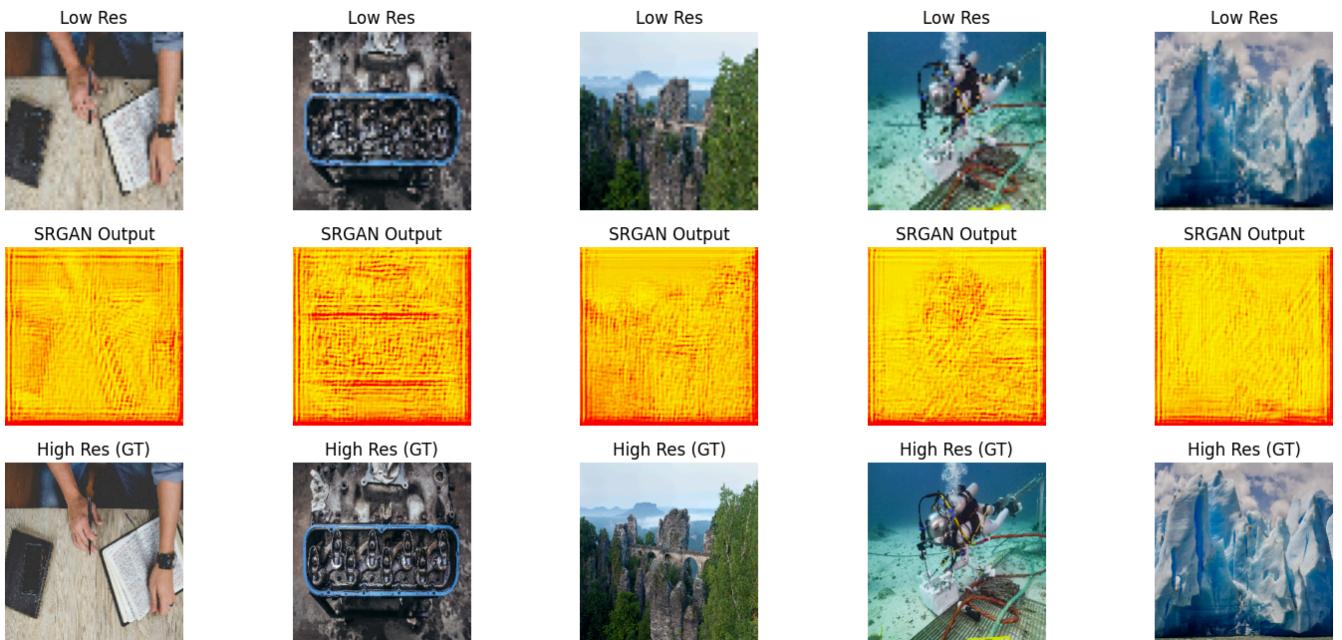
Epoch 1 | PSNR: 5.1355 | SSIM: 0.1112



[Epoch 01] G Loss: 0.0788 | D Loss: 2.0048 | PSNR: 5.1355 | SSIM: 0.1112

1/1 0s 166ms/step

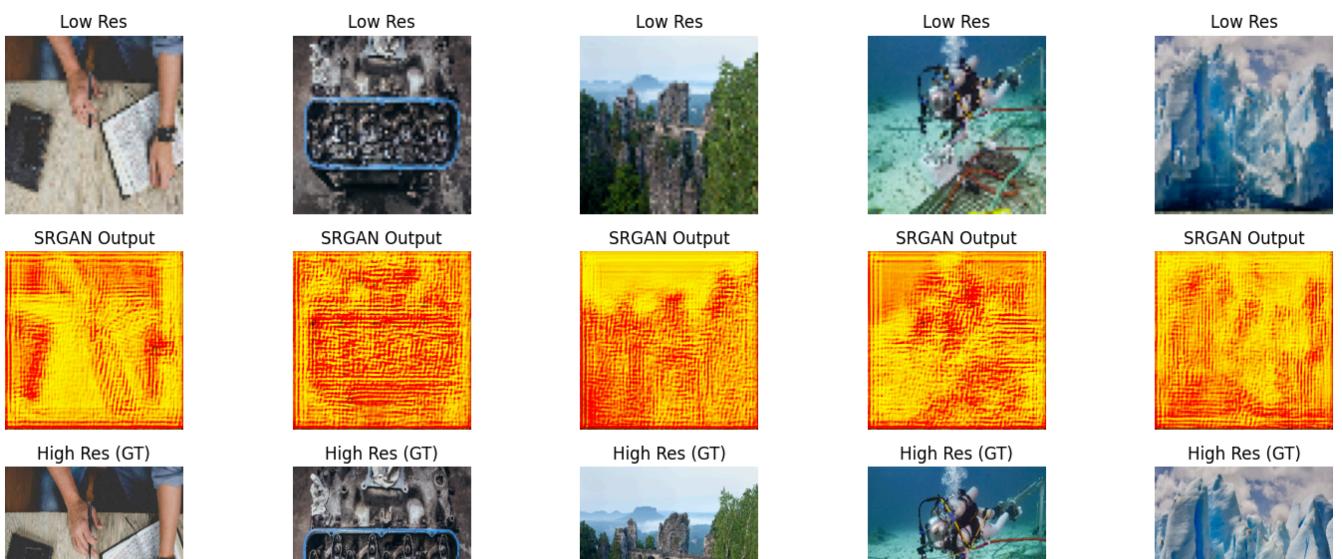
Epoch 2 | PSNR: 5.4153 | SSIM: 0.0889



[Epoch 02] G Loss: 0.0722 | D Loss: 1.4471 | PSNR: 5.4153 | SSIM: 0.0889

1/1 0s 154ms/step

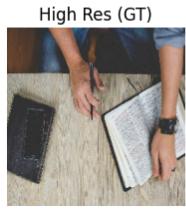
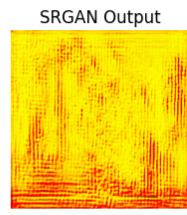
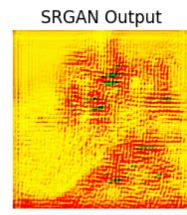
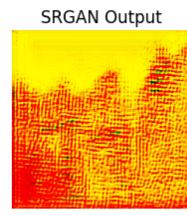
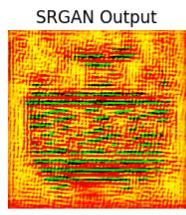
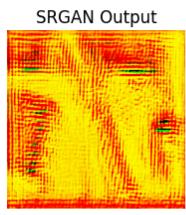
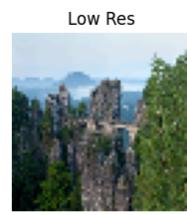
Epoch 3 | PSNR: 5.7035 | SSIM: 0.0741





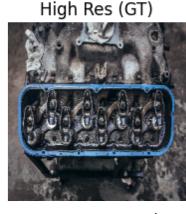
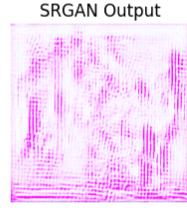
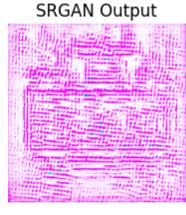
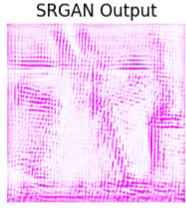
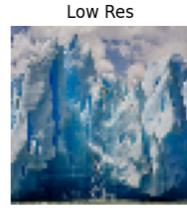
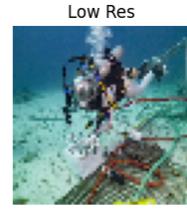
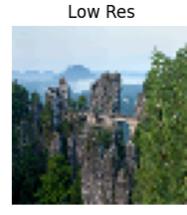
[Epoch 03] G Loss: 0.0694 | D Loss: 1.6379 | PSNR: 5.7035 | SSIM: 0.0741
1/1 0s 161ms/step

Epoch 4 | PSNR: 5.5690 | SSIM: 0.0771



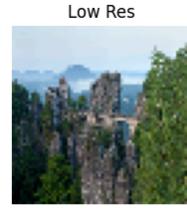
[Epoch 04] G Loss: 0.0675 | D Loss: 1.8745 | PSNR: 5.5690 | SSIM: 0.0771
1/1 0s 155ms/step

Epoch 5 | PSNR: 5.0146 | SSIM: 0.1710



[Epoch 05] G Loss: 0.0670 | D Loss: 2.1290 | PSNR: 5.0146 | SSIM: 0.1710
1/1 0s 160ms/step

Epoch 6 | PSNR: 5.3171 | SSIM: 0.1756



High Res (GT)

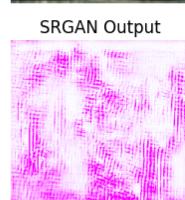
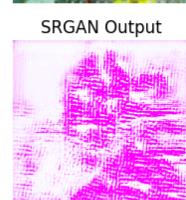
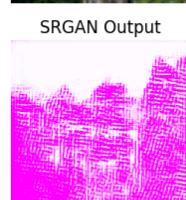
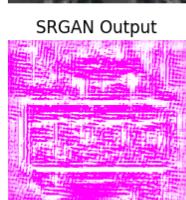
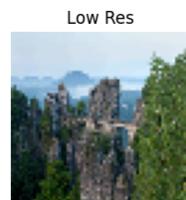
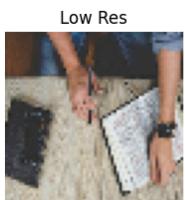
4/25/25, 5:50 PM

NN_Final_Submission (2).ipynb - Colab



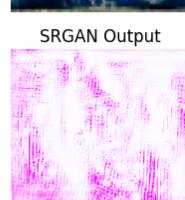
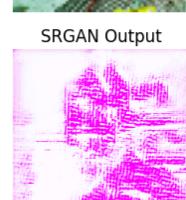
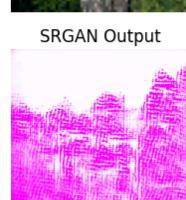
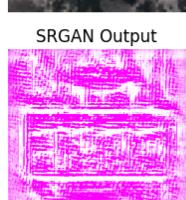
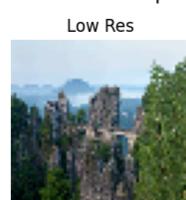
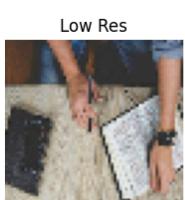
[Epoch 06] G Loss: 0.0632 | D Loss: 1.5816 | PSNR: 5.3171 | SSIM: 0.1756
1/1 0s 180ms/step

Epoch 7 | PSNR: 5.3021 | SSIM: 0.1701



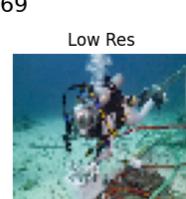
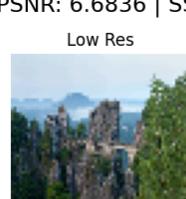
[Epoch 07] G Loss: 0.0588 | D Loss: 1.4742 | PSNR: 5.3021 | SSIM: 0.1701
1/1 0s 165ms/step

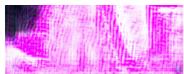
Epoch 8 | PSNR: 5.3169 | SSIM: 0.1799



[Epoch 08] G Loss: 0.0559 | D Loss: 1.4277 | PSNR: 5.3169 | SSIM: 0.1799
1/1 0s 156ms/step

Epoch 9 | PSNR: 6.6836 | SSIM: 0.1669





High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)

[Epoch 09] G Loss: 0.0535 | D Loss: 1.3068 | PSNR: 6.6836 | SSIM: 0.1669
1/1 0s 169ms/step

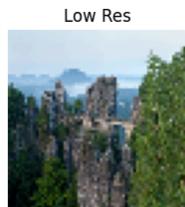
Epoch 10 | PSNR: 7.8967 | SSIM: 0.1769



Low Res



Low Res



Low Res



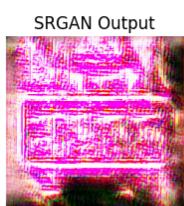
Low Res



Low Res



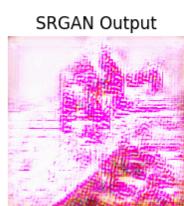
SRGAN Output



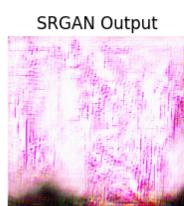
SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)

[Epoch 10] G Loss: 0.0520 | D Loss: 1.1881 | PSNR: 7.8967 | SSIM: 0.1769
1/1 0s 160ms/step

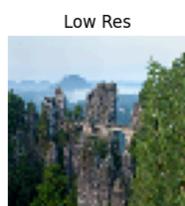
Epoch 11 | PSNR: 11.0715 | SSIM: 0.2238



Low Res



Low Res



Low Res



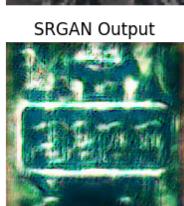
Low Res



Low Res



SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)

[Epoch 11] G Loss: 0.0519 | D Loss: 1.9713 | PSNR: 11.0715 | SSIM: 0.2238
1/1 0s 160ms/step

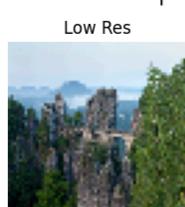
Epoch 12 | PSNR: 9.6241 | SSIM: 0.1598



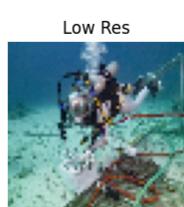
Low Res



Low Res



Low Res



Low Res



Low Res

SRGAN Output

SRGAN Output

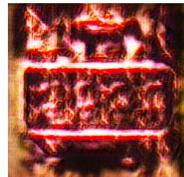
SRGAN Output

SRGAN Output

SRGAN Output



High Res (GT)



High Res (GT)



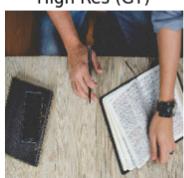
High Res (GT)



High Res (GT)



High Res (GT)



Low Res



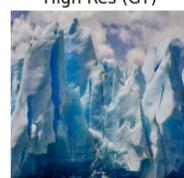
Low Res



Low Res



Low Res



Low Res

[Epoch 12] G Loss: 0.0523 | D Loss: 2.8976 | PSNR: 9.6241 | SSIM: 0.1598
1/1 0s 171ms/step

Epoch 13 | PSNR: 12.1896 | SSIM: 0.2719



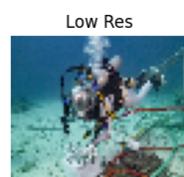
SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)



[Epoch 13] G Loss: 0.0476 | D Loss: 2.7214 | PSNR: 12.1896 | SSIM: 0.2719
1/1 0s 157ms/step

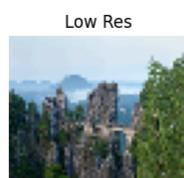
Epoch 14 | PSNR: 15.3176 | SSIM: 0.2821



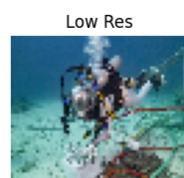
Low Res



Low Res



Low Res



Low Res



Low Res



SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



SRGAN Output



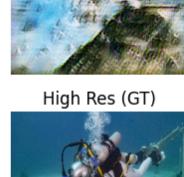
High Res (GT)



High Res (GT)



High Res (GT)



High Res (GT)

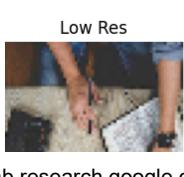


High Res (GT)



[Epoch 14] G Loss: 0.0432 | D Loss: 2.3473 | PSNR: 15.3176 | SSIM: 0.2821
1/1 0s 165ms/step

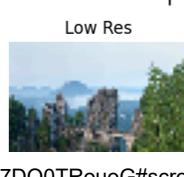
Epoch 15 | PSNR: 14.8218 | SSIM: 0.2974



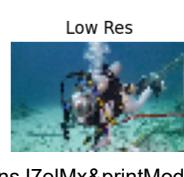
Low Res



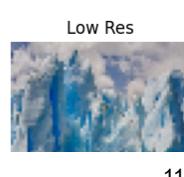
Low Res



Low Res



Low Res



Low Res



SRGAN Output



High Res (GT)



SRGAN Output

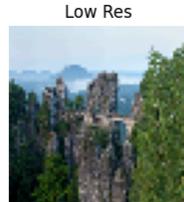


High Res (GT)



[Epoch 15] G Loss: 0.0411 | D Loss: 2.4764 | PSNR: 14.8218 | SSIM: 0.2974
1/1 0s 156ms/step

Epoch 16 | PSNR: 15.9284 | SSIM: 0.3184



Low Res



SRGAN Output



High Res (GT)



SRGAN Output



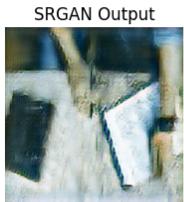
High Res (GT)



SRGAN Output

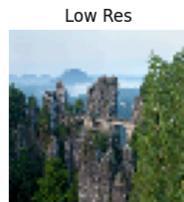


High Res (GT)



[Epoch 16] G Loss: 0.0391 | D Loss: 2.4419 | PSNR: 15.9284 | SSIM: 0.3184
1/1 0s 158ms/step

Epoch 17 | PSNR: 15.8292 | SSIM: 0.3141



Low Res



SRGAN Output



High Res (GT)



SRGAN Output



High Res (GT)



SRGAN Output



High Res (GT)



[Epoch 17] G Loss: 0.0377 | D Loss: 2.1444 | PSNR: 15.8292 | SSIM: 0.3141
1/1 0s 160ms/step

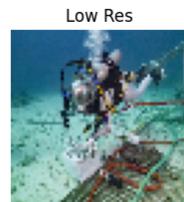
Epoch 18 | PSNR: 16.4110 | SSIM: 0.3264



Low Res

SRGAN Output

High Res (GT)



SRGAN Output



High Res (GT)



SRGAN Output



High Res (GT)

