```
# Install OpenCV for colorization
!pip install opencv-python-headless
!pip install torch torchvision
!pip install opencv-python numpy
!pip install ffmpeg
!pip install yt_dlp
!pip install scikit-image
!pip install tensorflow
!pip install lpips
!pip install streamlit pyngrok
!pip install gradio pillow
```

```
Requirement already satisfied: opencv-python-headless in
/usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in
/usr/local/lib/python3.10/dist-packages (from opencv-python-headless)
(1.26.4)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (2.5.1+cu121)
Requirement already satisfied: torchvision in
/usr/local/lib/python3.10/dist-packages (0.20.1+cu121)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch)
(1.3.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torchvision) (1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.10/dist-packages (from torchvision) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch) (2.1.5)
Requirement already satisfied: opencv-python in
/usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: ffmpeg in
/usr/local/lib/python3.10/dist-packages (1.4)
Requirement already satisfied: yt_dlp in
/usr/local/lib/python3.10/dist-packages (2024.11.18)
```

```
Requirement already satisfied: scikit-image in
/usr/local/lib/python3.10/dist-packages (0.24.0)
Requirement already satisfied: numpy>=1.23 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (1.26.4)
Requirement already satisfied: scipy>=1.9 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (1.13.1)
Requirement already satisfied: networkx>=2.8 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (3.4.2)
Requirement already satisfied: pillow>=9.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (11.0.0)
Requirement already satisfied: imageio>=2.33 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (2.36.0)
Requirement already satisfied: tifffile>=2022.8.12 in
/usr/local/lib/python3.10/dist-packages (from scikit-image)
(2024.9.20)
Requirement already satisfied: packaging>=21 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (24.2)
Requirement already satisfied: lazy-loader>=0.4 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (0.4)
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.10/dist-packages (2.17.1)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/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,<5.0.0dev,>=3.20.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.25.5)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.68.0)
Requirement already satisfied: tensorboard<2.18,>=2.17 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.1)
Requirement already satisfied: keras>=3.2.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.5.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.45.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-
packages (from keras>=3.2.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in
/usr/local/lib/python3.10/dist-packages (from keras>=3.2.0-
>tensorflow) (0.0.8)
Requirement already satisfied: optree in
/usr/local/lib/python3.10/dist-packages (from keras>=3.2.0-
>tensorflow) (0.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17-
>tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17-
>tensorflow) (3.1.3)
```

```
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.18,>=2.17->tensorflow) (2.1.5)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0-
>tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0-
>tensorflow) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0-
>rich->keras>=3.2.0->tensorflow) (0.1.2)
Requirement already satisfied: lpips in
/usr/local/lib/python3.10/dist-packages (0.1.4)
Requirement already satisfied: torch>=0.4.0 in
/usr/local/lib/python3.10/dist-packages (from lpips) (2.5.1+cu121)
Requirement already satisfied: torchvision>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from lpips) (0.20.1+cu121)
Requirement already satisfied: numpy>=1.14.3 in
/usr/local/lib/python3.10/dist-packages (from lpips) (1.26.4)
Requirement already satisfied: scipy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from lpips) (1.13.1)
Requirement already satisfied: tqdm>=4.28.1 in
/usr/local/lib/python3.10/dist-packages (from lpips) (4.66.6)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=0.4.0->lpips)
(3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=0.4.0->lpips)
(4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=0.4.0->lpips)
(3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=0.4.0->lpips)
(3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch>=0.4.0->lpips)
(2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch>=0.4.0->lpips)
(1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch>=0.4.0->lpips) (1.3.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.10/dist-packages (from torchvision>=0.2.1-
>lpips) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=0.4.0->lpips) (2.1.5)
Requirement already satisfied: streamlit in /usr/local/lib/python3.10/dist-packages (1.40.2)
Requirement already satisfied: pyngrok in /usr/local/lib/python3.10/dist-packages (7.2.1)
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.2.2)
Requirement already satisfied: blinker<2,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (1.9.0)
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (5.5.0)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (8.1.7)
Requirement already satisfied: numpy<3,>=1.23 in /usr/local/lib/python3.10/dist-packages (from streamlit) (1.26.4)
Requirement already satisfied: packaging<25,>=20 in /usr/local/lib/python3.10/dist-packages (from streamlit) (24.2)
Requirement already satisfied: pandas<3,>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.2.2)
Requirement already satisfied: pillow<12,>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (11.0.0)
Requirement already satisfied: protobuf<6,>=3.20 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.25.5)
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (17.0.0)
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.32.3)
Requirement already satisfied: rich<14,>=10.14.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (13.9.4)
Requirement already satisfied: tenacity<10,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (9.0.0)
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.12.2)
Requirement already satisfied: watchdog<7,>=2.1.5 in /usr/local/lib/python3.10/dist-packages (from streamlit) (6.0.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local/lib/python3.10/dist-packages (from streamlit) (3.1.43)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in /usr/local/lib/python3.10/dist-packages (from streamlit) (0.9.1)
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.10/dist-packages (from streamlit) (6.3.3)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.10/dist-packages (from pyngrok) (6.0.2)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.4)
```

```
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0-
>streamlit) (3.1.4)
Requirement already satisfied: jsonschema>=3.0 in
/usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0-
>streamlit) (4.23.0)
Requirement already satisfied: toolz in
/usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0-
>streamlit) (0.12.1)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/usr/local/lib/python3.10/dist-packages (from gitpython!
=3.1.19,<4,>=3.0.7->streamlit) (4.0.11)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.4.0-
>streamlit) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.4.0-
>streamlit) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.4.0-
>streamlit) (2024.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27-
>streamlit) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27-
>streamlit) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27-
>streamlit) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27-
>streamlit) (2024.8.30)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich<14,>=10.14.0-
>streamlit) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich<14,>=10.14.0-
>streamlit) (2.18.0)
Requirement already satisfied: smmap<6,>=3.0.1 in
/usr/local/lib/python3.10/dist-packages (from gitdb<5,>=4.0.1-
>gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.1)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->altair<6,>=4.0-
>streamlit) (2.1.5)
Requirement already satisfied: attrs>=22.2.0 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
```

```
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit) (2024.10.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit) (0.21.0)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0-
>rich<14,>=10.14.0->streamlit) (0.1.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas<3,>=1.4.0->streamlit) (1.16.0)
Requirement already satisfied: gradio in
/usr/local/lib/python3.10/dist-packages (5.7.1)
Requirement already satisfied: pillow in
/usr/local/lib/python3.10/dist-packages (11.0.0)
Requirement already satisfied: aiofiles<24.0,>=22.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (23.2.1)
Requirement already satisfied: anyio<5.0,>=3.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (3.7.1)
Requirement already satisfied: fastapi<1.0,>=0.115.2 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.115.5)
Requirement already satisfied: ffmpy in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.4.0)
Requirement already satisfied: gradio-client==1.5.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (1.5.0)
Requirement already satisfied: httpx>=0.24.1 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.27.2)
Requirement already satisfied: huggingface-hub>=0.25.1 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.26.2)
Requirement already satisfied: jinja2<4.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (3.1.4)
Requirement already satisfied: markupsafe~=2.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (2.1.5)
Requirement already satisfied: numpy<3.0,>=1.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (1.26.4)
Requirement already satisfied: orjson~=3.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (3.10.11)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pydantic>=2.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (2.9.2)
Requirement already satisfied: pydub in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.25.1)
Requirement already satisfied: python-multipart==0.0.12 in
```

```
/usr/local/lib/python3.10/dist-packages (from gradio) (0.0.12)
Requirement already satisfied: pyyaml<7.0,>=5.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (6.0.2)
Requirement already satisfied: ruff>=0.2.2 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.8.1)
Requirement already satisfied: safehttpx<1.0,>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.1.6)
Requirement already satisfied: semantic-version~=2.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (2.10.0)
Requirement already satisfied: starlette<1.0,>=0.40.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.41.3)
Requirement already satisfied: tomlkit==0.12.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.12.0)
Requirement already satisfied: typer<1.0,>=0.12 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.13.0)
Requirement already satisfied: typing-extensions~=4.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (4.12.2)
Requirement already satisfied: uvicorn>=0.14.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.32.1)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from gradio-client==1.5.0-
>gradio) (2024.10.0)
Requirement already satisfied: websockets<13.0,>=10.0 in
/usr/local/lib/python3.10/dist-packages (from gradio-client==1.5.0-
>gradio) (12.0)
Requirement already satisfied: idna>=2.8 in
/usr/local/lib/python3.10/dist-packages (from anyio<5.0,>=3.0->gradio)
(3.10)
Requirement already satisfied: sniffio>=1.1 in
/usr/local/lib/python3.10/dist-packages (from anyio<5.0,>=3.0->gradio)
(1.3.1)
Requirement already satisfied: exceptiongroup in
/usr/local/lib/python3.10/dist-packages (from anyio<5.0,>=3.0->gradio)
(1.2.2)
Requirement already satisfied: certifi in
/usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio)
(2024.8.30)
Requirement already satisfied: httpcore==1.* in
/usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio)
(1.0.7)
Requirement already satisfied: h11<0.15,>=0.13 in
/usr/local/lib/python3.10/dist-packages (from httpcore==1.*-
>httpx>=0.24.1->gradio) (0.14.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.25.1-
>gradio) (3.16.1)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.25.1-
>gradio) (2.32.3)
```

```
Requirement already satisfied: tqdm>=4.42.1 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.25.1-
>gradio) (4.66.6)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0-
>gradio) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0-
>gradio) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0-
>gradio) (2024.2)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio)
(0.7.0)
Requirement already satisfied: pydantic-core==2.23.4 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio)
(2.23.4)
Requirement already satisfied: click>=8.0.0 in
/usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12-
>gradio) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in
/usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12-
>gradio) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in
/usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12-
>gradio) (13.9.4)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas<3.0,>=1.0->gradio) (1.16.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich>=10.11.0-
>typer<1.0,>=0.12->gradio) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich>=10.11.0-
>typer<1.0,>=0.12->gradio) (2.18.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.25.1->gradio) (3.4.0)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.25.1->gradio) (2.2.3)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0-
>rich>=10.11.0->typer<1.0,>=0.12->gradio) (0.1.2)

import zipfile
import os

# Unzip the uploaded dataset
```

```python
zip_path = '/content/DataSET.zip'  # Path to your uploaded zip file
extract_path = '/content/dataset'  # Target extraction directory

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

# Verify the dataset structure
print("Dataset Structure:")
for root, dirs, files in os.walk(extract_path):
    level = root.replace(extract_path, '').count(os.sep)
    indent = ' ' * 4 * level
    print(f"{indent}{os.path.basename(root)}/")
    sub_indent = ' ' * 4 * (level + 1)
    for file in files:
        print(f"{sub_indent}{file}")
```

```
Dataset Structure:
dataset/
    colorization_model.pth
    ColorizeArtistic_gen_saved.pth
    DataSET/
        Colorized/
            image5018.jpg
            image5011.jpg
            image5014.jpg
            image5012.jpg
            image5009.jpg
            image5003.jpg
            image5021.jpg
            image5015.jpg
            image5007.jpg
            image5006.jpg
            image5010.jpg
            image5000.jpg
            image5013.jpg
            image5019.jpg
            image5020.jpg
            image5005.jpg
            image5001.jpg
            image5004.jpg
            image5017.jpg
            image5016.jpg
            image5002.jpg
            image5008.jpg
        Gray/
            image5018.jpg
            image5011.jpg
            image5014.jpg
            image5012.jpg
            image5009.jpg
```

```
            image5003.jpg
            image5021.jpg
            image5015.jpg
            image5007.jpg
            image5006.jpg
            image5010.jpg
            image5000.jpg
            image5013.jpg
            image5019.jpg
            image5020.jpg
            image5005.jpg
            image5001.jpg
            image5004.jpg
            image5017.jpg
            image5016.jpg
            image5002.jpg
            image5008.jpg
        Orignal/
            image5018.jpg
            image5011.jpg
            image5014.jpg
            image5012.jpg
            image5009.jpg
            image5003.jpg
            image5021.jpg
            image5015.jpg
            image5007.jpg
            image5006.jpg
            image5010.jpg
            image5000.jpg
            image5013.jpg
            image5019.jpg
            image5020.jpg
            image5005.jpg
            image5001.jpg
            image5004.jpg
            image5017.jpg
            image5016.jpg
            image5002.jpg
            image5008.jpg
        Output/
            image5018.jpg
            image5011.jpg
            image5014.jpg
            image5012.jpg
            image5009.jpg
            image5003.jpg
            image5021.jpg
            image5015.jpg
```

```
image5007.jpg
image5006.jpg
image5010.jpg
image5000.jpg
image5013.jpg
image5019.jpg
image5020.jpg
image5005.jpg
image5001.jpg
image5004.jpg
image5017.jpg
image5016.jpg
image5002.jpg
image5008.jpg
```

```
import os
from PIL import Image

# Define dataset paths
gray_images_path = "/content/dataset/DataSET/Gray"
original_images_path = "/content/dataset/DataSET/Orignal"
output_path = "/content/colorized_outputs"
os.makedirs(output_path, exist_ok=True)

# List of all grayscale images
gray_image_files = sorted([f for f in os.listdir(gray_images_path) if
f.endswith(('.jpg', '.png', '.jpeg'))])
original_image_files = sorted([f for f in
os.listdir(original_images_path) if f.endswith(('.jpg', '.png',
'.jpeg'))])

assert len(gray_image_files) == len(original_image_files), "Mismatch
in gray and original image counts."
print(f"Found {len(gray_image_files)} grayscale and original images.")
```

```
Found 22 grayscale and original images.
```

```
import os
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

# Set dataset path
dataset_path = '/content/dataset/DataSET'

# Load images
gray_images = []
color_images = []

# Load grayscale images and their corresponding color images
```

```
for file in os.listdir(os.path.join(dataset_path, 'Gray')):
    if file.endswith('.png') or file.endswith('.jpg'):
        gray_image = cv2.imread(os.path.join(dataset_path, 'Gray',
file), cv2.IMREAD_GRAYSCALE)
        color_image = cv2.imread(os.path.join(dataset_path, 'Orignal',
file))

        gray_images.append(gray_image)
        color_images.append(color_image)

# Convert lists to numpy arrays for processing
gray_images = np.array(gray_images)
color_images = np.array(color_images)

# Clone the DeOldify repository and set up the environment
!git clone https://github.com/jantic/DeOldify.git DeOldify
%cd DeOldify

# Fix for collections issue
import collections
import collections.abc
collections.Sized = collections.abc.Sized
collections.Mapping = collections.abc.Mapping
collections.MutableMapping = collections.abc.MutableMapping

# Set up the device
from deoldify import device
from deoldify.device_id import DeviceId
device.set(device=DeviceId.GPU0)

import torch
if not torch.cuda.is_available():
    print('GPU not available.')

# Install required dependencies
!pip install fastai==1.0.61 matplotlib opencv-python pillow

# Import DeOldify libraries
from deoldify.visualize import *
import warnings
warnings.filterwarnings("ignore", category=UserWarning, message=".*?
Your .*? set is empty.*?")

# Download the pre-trained model
!mkdir 'models'
!wget https://data.deepai.org/deoldify/ColorizeArtistic_gen.pth -
O ./models/ColorizeArtistic_gen.pth
colorizer = get_image_colorizer(artistic=True)

# Dataset paths
```

```
gray_images_path = '/content/dataset/DataSET/Gray'
original_images_path = '/content/dataset/DataSET/Orignal'
output_path = '/content/dataset/DataSET/Colorized'  # Path to save
colorized images

# Create output directory
import os
os.makedirs(output_path, exist_ok=True)

# Helper function to process and colorize images
def colorize_images_from_folder(input_folder, output_folder,
render_factor=35, watermarked=True):
    from PIL import Image
    images = [f for f in os.listdir(input_folder) if
f.lower().endswith(('png', 'jpg', 'jpeg'))]
    for image_file in images:
        input_path = os.path.join(input_folder, image_file)
        try:
            print(f"Processing: {input_path}")
            colorized_image_path = colorizer.plot_transformed_image(
                path=input_path,
                render_factor=render_factor,
                watermarked=watermarked,
                figsize=(8, 8)
            )
            # Save the colorized image
            output_file_path = os.path.join(output_folder, image_file)
            Image.open(colorized_image_path).save(output_file_path)
        except Exception as e:
            print(f"Error processing {image_file}: {e}")

# Colorize all images in the gray images folder
colorize_images_from_folder(gray_images_path, output_path,
render_factor=35)

# Import evaluation metrics
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim
import numpy as np
from PIL import Image

def evaluate_images(original_image_path, colorized_image_path):
    original_image =
np.array(Image.open(original_image_path).convert('RGB'))
    colorized_image =
np.array(Image.open(colorized_image_path).convert('RGB'))

    # Resize images to the same size if needed
    if original_image.shape != colorized_image.shape:
        colorized_image =
```

```
np.array(Image.open(colorized_image_path).resize(original_image.shape[
:2][::-1]))

    # Compute metrics
    psnr_value = psnr(original_image, colorized_image,
data_range=colorized_image.max() - colorized_image.min())

    # Adjust `win_size` for small images
    min_dim = min(original_image.shape[0], original_image.shape[1])
    win_size = 7 if min_dim >= 7 else min_dim // 2 * 2 + 1  # Ensure
win_size is odd and <= min_dim

    ssim_value = ssim(
        original_image,
        colorized_image,
        channel_axis=-1,
        win_size=win_size
    )

    mse_value = np.mean((original_image - colorized_image) ** 2)

    return psnr_value, ssim_value, mse_value

# Evaluate each image and print metrics
for image_file in os.listdir(gray_images_path):
    original_image_path = os.path.join(original_images_path,
image_file)
    colorized_image_path = os.path.join(output_path, image_file)
    try:
        psnr_value, ssim_value, mse_value =
evaluate_images(original_image_path, colorized_image_path)
        print(f"Image: {image_file}")
        print(f"  PSNR: {psnr_value:.2f}")
        print(f"  SSIM: {ssim_value:.4f}")
        print(f"  MSE: {mse_value:.2f}")
    except Exception as e:
        print(f"Error evaluating {image_file}: {e}")

# Backup the Pre-Trained Model
import shutil
# Copy the model file to a backup location
backup_path = '/content/dataset/ColorizeArtistic_gen_saved.pth'
shutil.copy('./models/ColorizeArtistic_gen.pth', backup_path)
print(f"Model backup created successfully at {backup_path}!")

Cloning into 'DeOldify'...
remote: Enumerating objects: 2620, done.ote: Counting objects: 100%
(274/274), done.ote: Compressing objects: 100% (194/194), done.ote:
Total 2620 (delta 93), reused 212 (delta 73), pack-reused 2346 (from
1)ent already satisfied: fastai==1.0.61 in
```

```
/usr/local/lib/python3.10/dist-packages (1.0.61)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.8.0)
Requirement already satisfied: opencv-python in
/usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: pillow in
/usr/local/lib/python3.10/dist-packages (11.0.0)
Requirement already satisfied: bottleneck in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (1.4.2)
Requirement already satisfied: fastprogress>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (1.0.3)
Requirement already satisfied: beautifulsoup4 in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (4.12.3)
Requirement already satisfied: numexpr in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (2.10.1)
Requirement already satisfied: numpy>=1.15 in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (1.26.4)
Requirement already satisfied: nvidia-ml-py3 in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61)
(7.352.0)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (2.2.2)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (24.2)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (6.0.2)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (2.32.3)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61) (1.13.1)
Requirement already satisfied: torch>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61)
(2.5.1+cu121)
Requirement already satisfied: torchvision in
/usr/local/lib/python3.10/dist-packages (from fastai==1.0.61)
(0.20.1+cu121)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.55.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
```

```
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=1.0.0-
>fastai==1.0.61) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.0.0-
>fastai==1.0.61) (4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=1.0.0-
>fastai==1.0.61) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.0.0-
>fastai==1.0.61) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch>=1.0.0-
>fastai==1.0.61) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.0.0-
>fastai==1.0.61) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch>=1.0.0->fastai==1.0.61) (1.3.0)
Requirement already satisfied: soupsieve>1.2 in
/usr/local/lib/python3.10/dist-packages (from beautifulsoup4-
>fastai==1.0.61) (2.6)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas->fastai==1.0.61)
(2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas->fastai==1.0.61)
(2024.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests-
>fastai==1.0.61) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests-
>fastai==1.0.61) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests-
>fastai==1.0.61) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests-
>fastai==1.0.61) (2024.8.30)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.0.0-
>fastai==1.0.61) (2.1.5)
mkdir: cannot create directory ‘models’: File exists
--2024-12-03 20:39:10--
```

```
https://data.deepai.org/deoldify/ColorizeArtistic_gen.pth
Resolving data.deepai.org (data.deepai.org)... 169.150.249.164,
2400:52e0:1a01::997:1
Connecting to data.deepai.org (data.deepai.org)|
169.150.249.164|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 255144681 (243M) [application/octet-stream]
Saving to: ‘./models/ColorizeArtistic_gen.pth’

./models/ColorizeAr 100%[===================>] 243.32M  5.11MB/s    in
50s

2024-12-03 20:40:00 (4.91 MB/s) - ‘./models/ColorizeArtistic_gen.pth’
saved [255144681/255144681]


/usr/local/lib/python3.10/dist-packages/torch/utils/data/
dataloader.py:617: UserWarning: This DataLoader will create 8 worker
processes in total. Our suggested max number of worker in current
system is 2, which is smaller than what this DataLoader is going to
create. Please be aware that excessive worker creation might get
DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:2
08: UserWarning: The parameter 'pretrained' is deprecated since 0.13
and may be removed in the future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:2
23: UserWarning: Arguments other than a weight enum or `None` for
'weights' are deprecated since 0.13 and may be removed in the future.
The current behavior is equivalent to passing
`weights=ResNet34_Weights.IMAGENET1K_V1`. You can also use
`weights=ResNet34_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
/usr/local/lib/python3.10/dist-packages/torch/nn/utils/weight_norm.py:
143: FutureWarning: `torch.nn.utils.weight_norm` is deprecated in
favor of `torch.nn.utils.parametrizations.weight_norm`.
  WeightNorm.apply(module, name, dim)
/content/DeOldify/fastai/basic_train.py:322: FutureWarning: You are
using `torch.load` with `weights_only=False` (the current default
value), which uses the default pickle module implicitly. It is
possible to construct malicious pickle data which will execute
arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
```

```
`torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
  state = torch.load(tmp_file)
/content/DeOldify/fastai/basic_train.py:271: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
  state = torch.load(source, map_location=device)

Processing: /content/dataset/DataSET/Gray/image5018.jpg
Processing: /content/dataset/DataSET/Gray/image5011.jpg
Processing: /content/dataset/DataSET/Gray/image5014.jpg
Processing: /content/dataset/DataSET/Gray/image5012.jpg
Processing: /content/dataset/DataSET/Gray/image5009.jpg
Processing: /content/dataset/DataSET/Gray/image5003.jpg
Processing: /content/dataset/DataSET/Gray/image5021.jpg
Processing: /content/dataset/DataSET/Gray/image5015.jpg
Processing: /content/dataset/DataSET/Gray/image5007.jpg
Processing: /content/dataset/DataSET/Gray/image5006.jpg
Processing: /content/dataset/DataSET/Gray/image5010.jpg
Processing: /content/dataset/DataSET/Gray/image5000.jpg
Processing: /content/dataset/DataSET/Gray/image5013.jpg
Processing: /content/dataset/DataSET/Gray/image5019.jpg
Processing: /content/dataset/DataSET/Gray/image5020.jpg
Processing: /content/dataset/DataSET/Gray/image5005.jpg
Processing: /content/dataset/DataSET/Gray/image5001.jpg
Processing: /content/dataset/DataSET/Gray/image5004.jpg
Processing: /content/dataset/DataSET/Gray/image5017.jpg
Processing: /content/dataset/DataSET/Gray/image5016.jpg
Processing: /content/dataset/DataSET/Gray/image5002.jpg

/content/DeOldify/deoldify/visualize.py:152: RuntimeWarning: More than 20 figures have been opened. Figures created through the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicitly closed and may consume too much memory. (To control this warning, see the rcParam `figure.max_open_warning`). Consider using
```

```
`matplotlib.pyplot.close()`.
  fig, axes = plt.subplots(1, 1, figsize=figsize)

Processing: /content/dataset/DataSET/Gray/image5008.jpg
Image: image5018.jpg
  PSNR: 28.40
  SSIM: 0.9715
  MSE: 42.51
Image: image5011.jpg
  PSNR: 19.87
  SSIM: 0.8130
  MSE: 74.17
Image: image5014.jpg
  PSNR: 24.14
  SSIM: 0.9334
  MSE: 51.41
Image: image5012.jpg
  PSNR: 23.09
  SSIM: 0.8749
  MSE: 55.11
Image: image5009.jpg
  PSNR: 17.94
  SSIM: 0.8727
  MSE: 89.55
Image: image5003.jpg
  PSNR: 25.04
  SSIM: 0.9274
  MSE: 58.32
Image: image5021.jpg
  PSNR: 23.67
  SSIM: 0.9272
  MSE: 51.79
Image: image5015.jpg
  PSNR: 18.45
  SSIM: 0.9116
  MSE: 70.57
Image: image5007.jpg
  PSNR: 21.99
  SSIM: 0.8917
  MSE: 71.26
Image: image5006.jpg
  PSNR: 20.28
  SSIM: 0.9169
  MSE: 45.87
Image: image5010.jpg
  PSNR: 26.91
  SSIM: 0.9485
  MSE: 45.17
Image: image5000.jpg
  PSNR: 22.38
```

```
  SSIM: 0.9193
  MSE: 75.81
Image: image5013.jpg
  PSNR: 30.02
  SSIM: 0.9590
  MSE: 37.48
Image: image5019.jpg
  PSNR: 19.34
  SSIM: 0.9127
  MSE: 69.68
Image: image5020.jpg
  PSNR: 17.29
  SSIM: 0.8352
  MSE: 94.22
Image: image5005.jpg
  PSNR: 20.26
  SSIM: 0.8365
  MSE: 85.42
Image: image5001.jpg
  PSNR: 18.50
  SSIM: 0.8330
  MSE: 74.42
Image: image5004.jpg
  PSNR: 21.08
  SSIM: 0.8969
  MSE: 70.71
Image: image5017.jpg
  PSNR: 25.90
  SSIM: 0.9701
  MSE: 60.45
Image: image5016.jpg
  PSNR: 16.03
  SSIM: 0.7768
  MSE: 97.50
Image: image5002.jpg
  PSNR: 25.62
  SSIM: 0.9499
  MSE: 59.96
Image: image5008.jpg
  PSNR: 22.15
  SSIM: 0.8920
  MSE: 63.73
Model backup created successfully at
/content/dataset/ColorizeArtistic_gen_saved.pth!
```


DRINK
Coca-Cola
DELICIOUS AND REFRESHING
Sales Advertising
1038


DnB NOR
LUKOIL
adidas

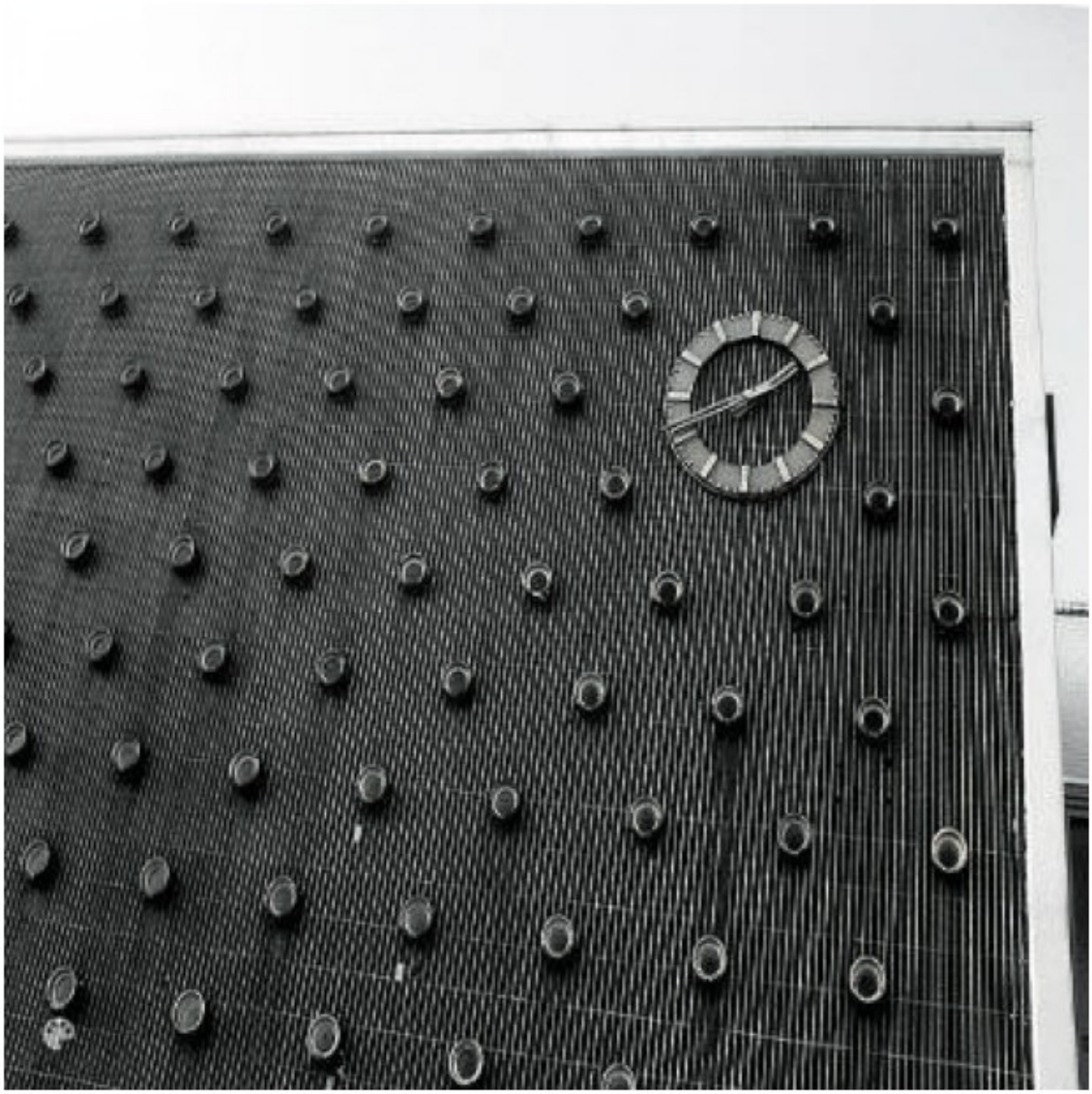
Rockwell International
Space Division

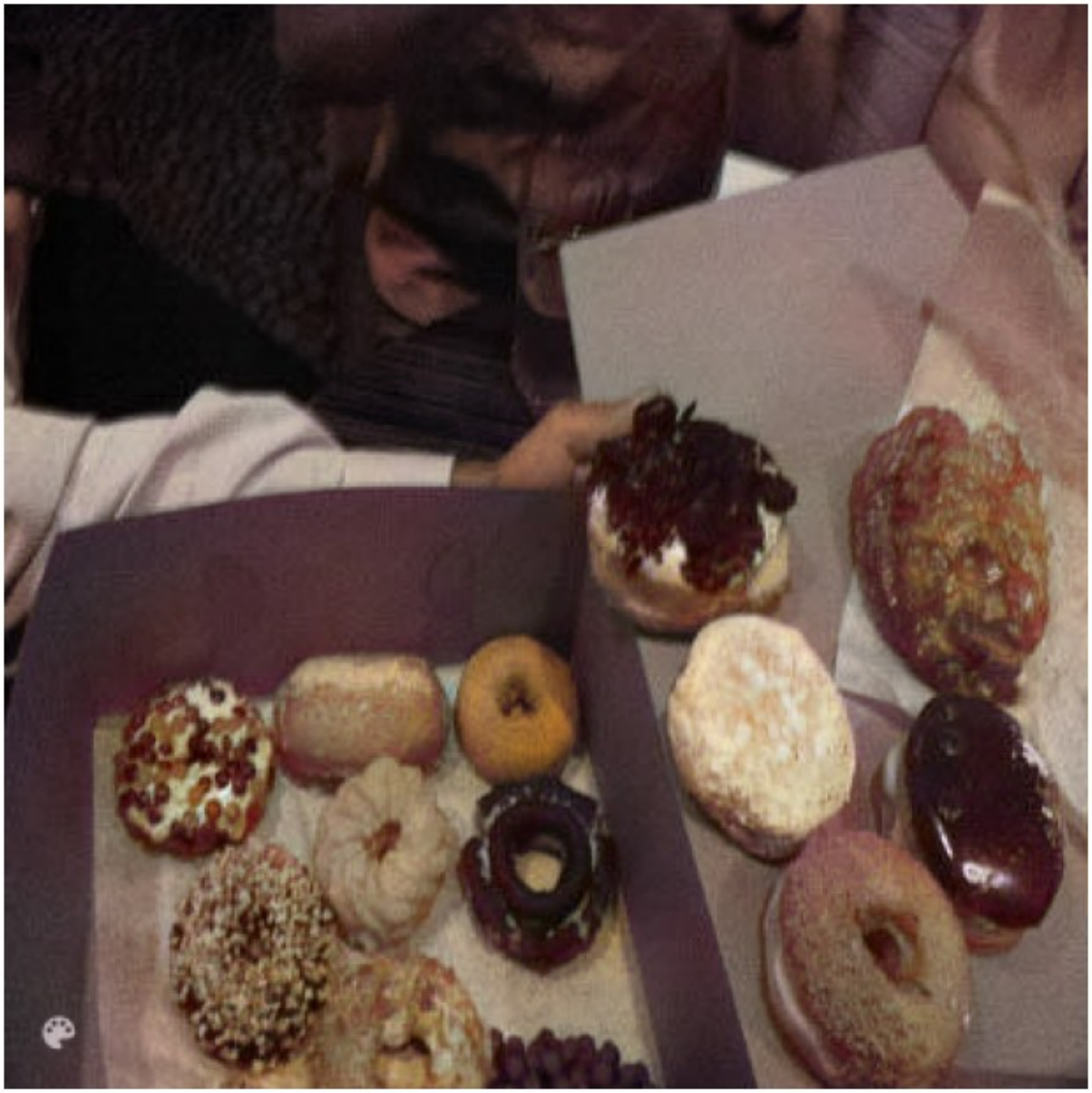

INTERNATIONAL
TOURNAMENT
LEPICA
MAN AND WOMAN
04/07/2012 08:03 PM

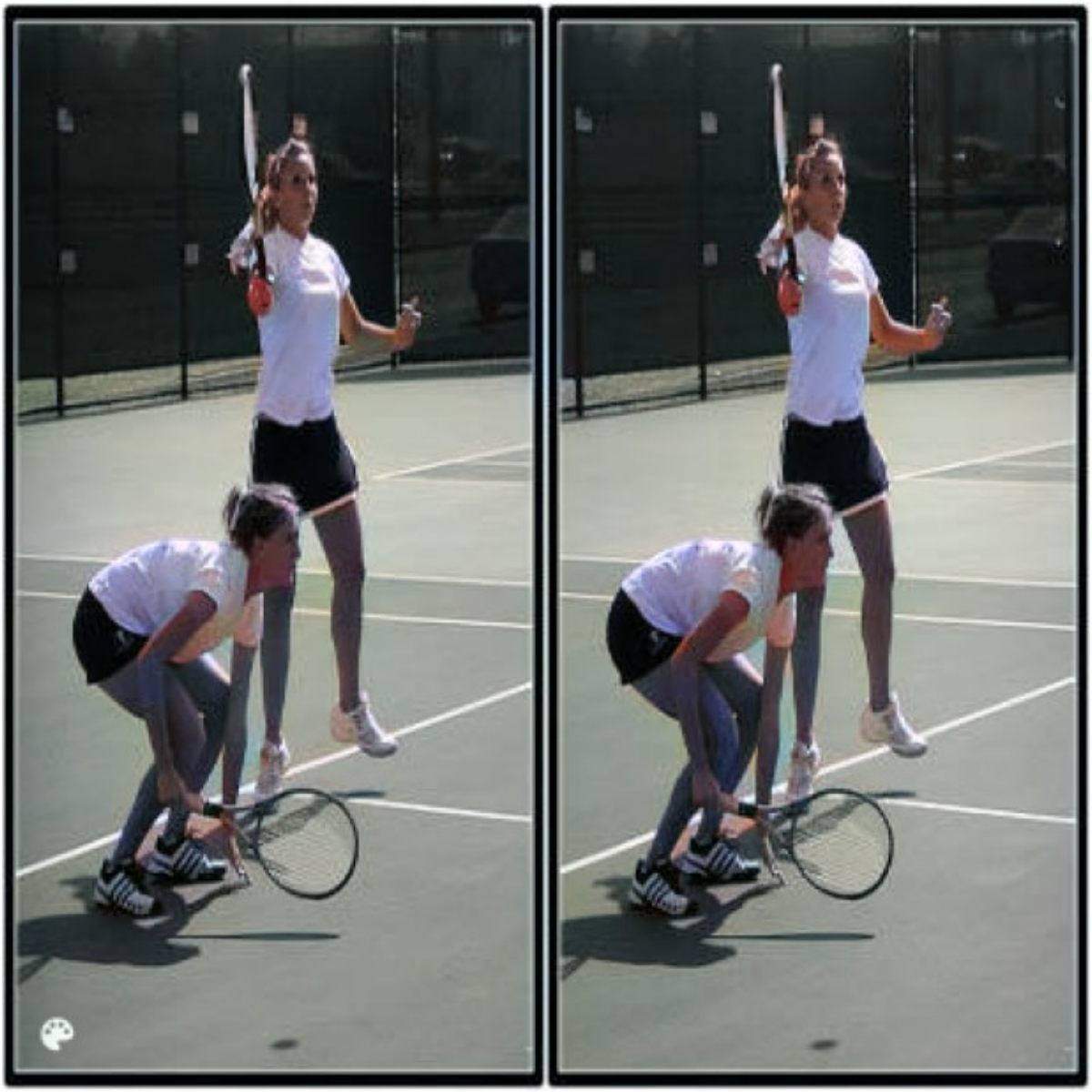

monste

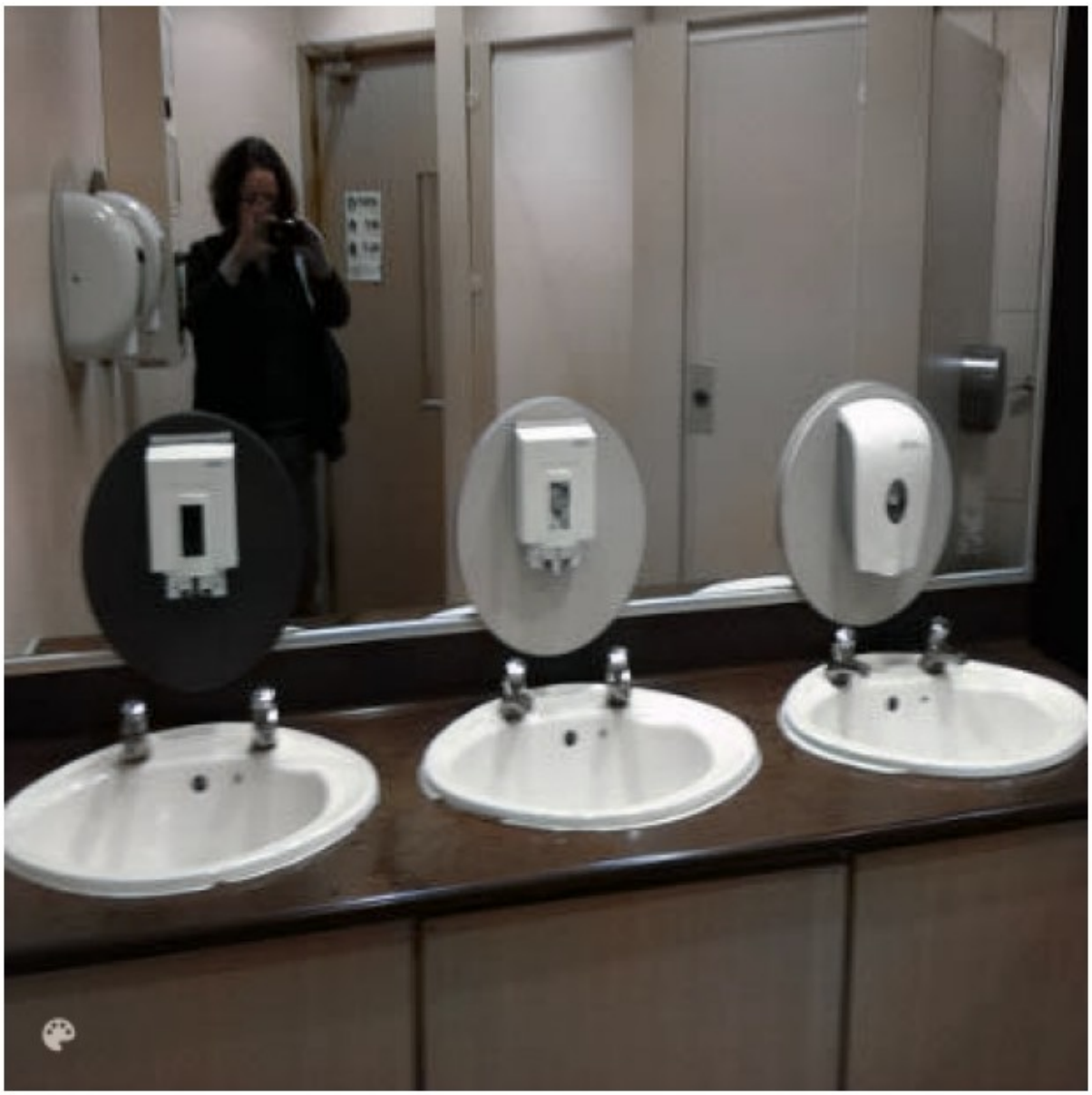

STO


Gelate









```
import os
import matplotlib.pyplot as plt
from PIL import Image

# Define your paths (update these variables with the correct paths)
gray_images_path = "/content/dataset/DataSET/Gray"
original_images_path = "/content/dataset/DataSET/Orignal"
output_path = "/content/dataset/DataSET/Colorized"

def display_images_side_by_side(original_image_path,
colorized_image_path):
    try:
        original_image = Image.open(original_image_path)
```

```
        colorized_image = Image.open(colorized_image_path)

        fig, axes = plt.subplots(1, 2, figsize=(12, 6))
        axes[0].imshow(original_image)
        axes[0].set_title('Original Image')
        axes[0].axis('off')

        axes[1].imshow(colorized_image)
        axes[1].set_title('Colorized Image')
        axes[1].axis('off')

        plt.show()
    except FileNotFoundError as e:
        print(f"Error: {e}")
        print(f"Original path: {original_image_path}")
        print(f"Colorized path: {colorized_image_path}")

# Compare the first few images (change range as needed)
for image_file in os.listdir(gray_images_path)[:5]:  # Adjust range as
needed
    gray_image_path = os.path.join(gray_images_path, image_file)
    original_image_path = os.path.join(original_images_path,
image_file)
    colorized_image_path = os.path.join(output_path, image_file)

    if os.path.exists(original_image_path) and
os.path.exists(colorized_image_path):
        display_images_side_by_side(original_image_path,
colorized_image_path)
    else:
        print(f"File missing for: {image_file}")
```

Original Image

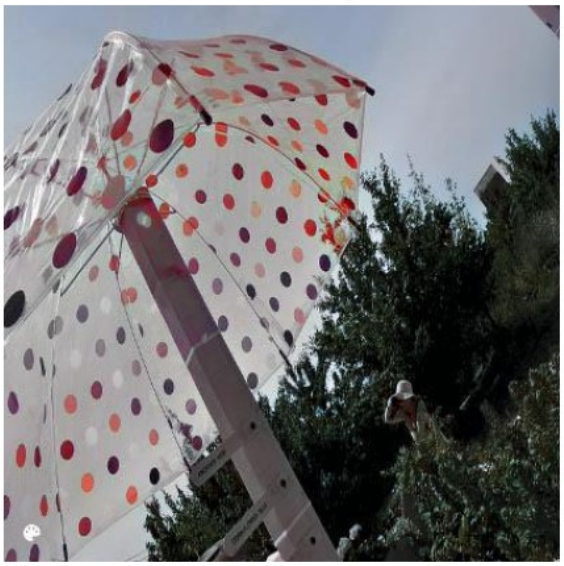
Colorized Image

Original Image

Colorized Image

Original Image

Colorized Image



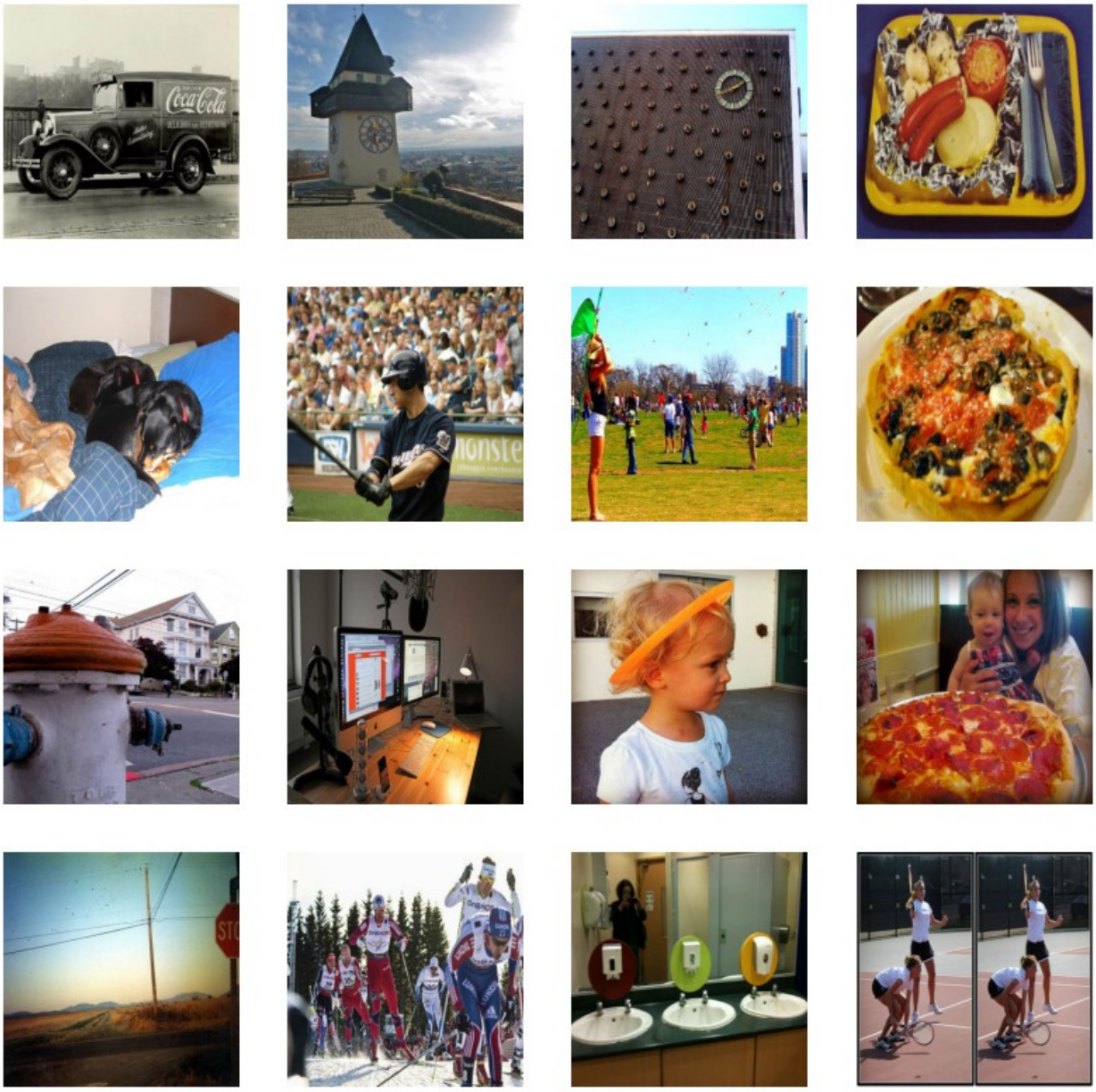
Original Image

Colorized Image

```
import os
import glob
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from pathlib import Path

# Define dataset paths
original_path = "/content/dataset/DataSET/Orignal"

# Load all image paths
paths = glob.glob(os.path.join(original_path, "*.jpg"))
```

```
print(f"Total images found: {len(paths)}")

# Dynamically adjust the sample size
total_images = len(paths)
sample_size = min(10_000, total_images)  # Ensure we don't sample more
than the total images

np.random.seed(123)
paths_subset = np.random.choice(paths, sample_size, replace=False)  #
Randomly sample images
rand_idxs = np.random.permutation(sample_size)

# Split into training and validation sets
train_idxs = rand_idxs[:int(sample_size * 0.8)]  # 80% for training
val_idxs = rand_idxs[int(sample_size * 0.8):]    # 20% for validation
train_paths = paths_subset[train_idxs]
val_paths = paths_subset[val_idxs]

print(f"Training samples: {len(train_paths)}, Validation samples:
{len(val_paths)}")

# Visualize some training images
_, axes = plt.subplots(4, 4, figsize=(10, 10))
for ax, img_path in zip(axes.flatten(), train_paths[:16]):  # Show up
to 16 images
    ax.imshow(Image.open(img_path))
    ax.axis("off")
plt.show()
```

```
Total images found: 22
Training samples: 17, Validation samples: 5
```




```
import cv2
import os
import numpy as np
import torch
import torch.nn as nn
from skimage.metrics import structural_similarity as ssim
from google.colab.patches import cv2_imshow  # For displaying images
in Colab

# Define the input and output paths
gray_folder = '/content/dataset/DataSET/Gray'
original_folder = '/content/dataset/DataSET/Original'
```

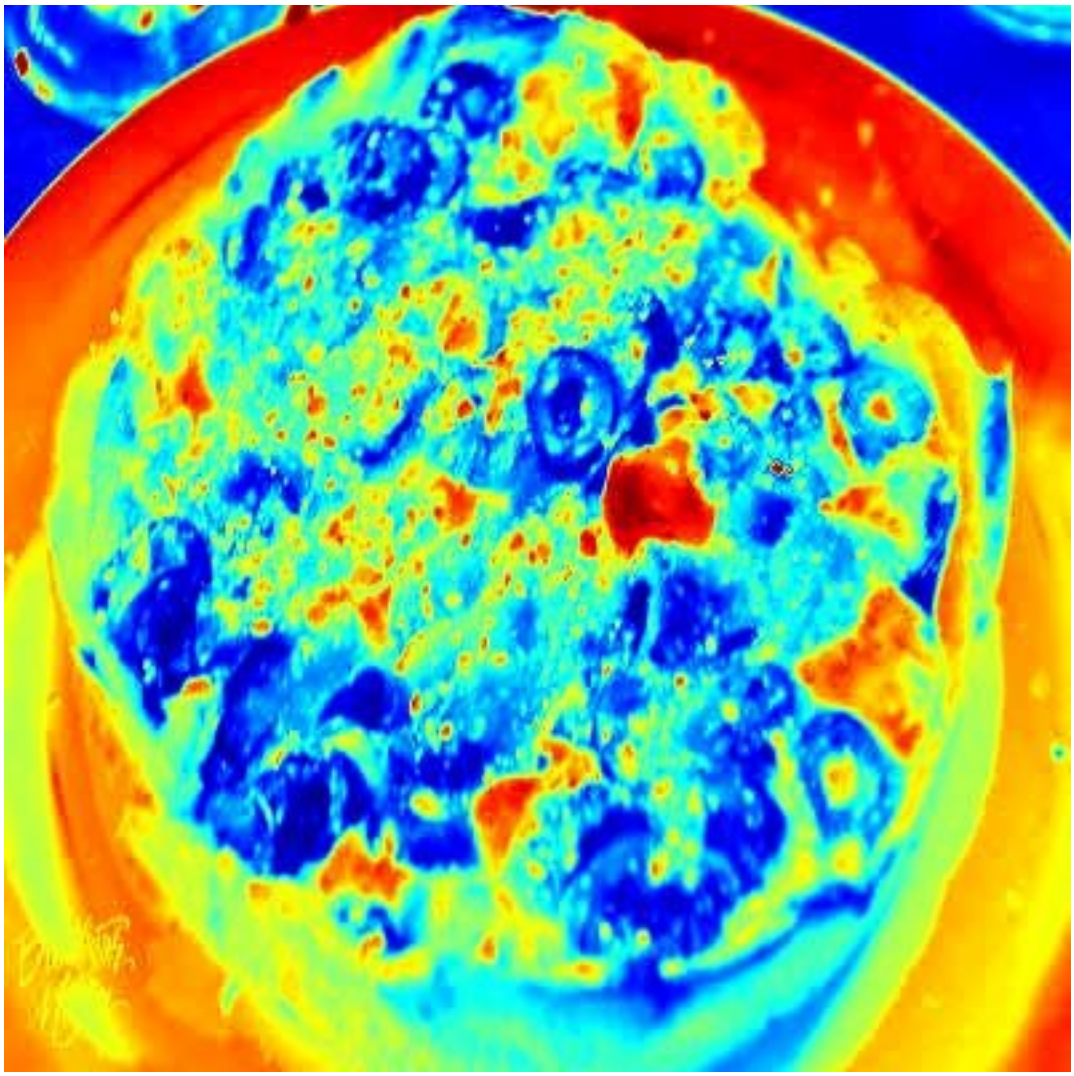
```
output_folder = '/content/dataset/DataSET/Output'  # New folder for
output images
model_path = '/content/dataset/colorization_model.pth'  # Path to save
the model

# Create the output folder if it doesn't exist
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

# Dummy PyTorch model for demonstration
class ColorizationModelClass(torch.nn.Module):
    def _init_(self):
        super(ColorizationModelClass, self)._init_()
        self.conv1 = torch.nn.Conv2d(1, 64, kernel_size=3, stride=1,
padding=1)  # Grayscale input
        self.conv2 = torch.nn.Conv2d(64, 128, kernel_size=3, stride=1,
padding=1)
        self.conv3 = torch.nn.Conv2d(128, 3, kernel_size=3, stride=1,
padding=1)  # RGB output

    def forward(self, x):
        x = torch.relu(self.conv1(x))
        x = torch.relu(self.conv2(x))
        x = torch.sigmoid(self.conv3(x))
        return x


# Instantiate and save the model
def save_model():
    model = ColorizationModelClass()
    torch.save(model.state_dict(), model_path)
    print(f"Model saved at {model_path}")

# Load the model and its weights
def load_model(model_path):
    model = ColorizationModelClass()
    model.load_state_dict(torch.load(model_path,
map_location=torch.device('cpu')))
    model.eval()
    return model

# Function to compute evaluation metrics
def compute_metrics(gray_image, colorized_image):
    # Convert colorized image to grayscale for comparison
    colorized_gray = cv2.cvtColor(colorized_image, cv2.COLOR_BGR2GRAY)

    # Compute MSE
    mse_value = np.mean((gray_image - colorized_gray) ** 2)

    # Compute PSNR (avoid division by zero)
```

```
    psnr_value = cv2.PSNR(gray_image, colorized_gray)

    # Compute SSIM
    ssim_value, _ = ssim(gray_image, colorized_gray, full=True)

    return mse_value, psnr_value, ssim_value

# Function to colorize an image using OpenCV colormap
def colorize_image(image_path, output_path, target_size):
    # Load the grayscale image
    gray_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Check if the image is loaded successfully
    if gray_image is None:
        print(f"Error loading image: {image_path}")
        return None, None

    # Resize grayscale image to the target size
    gray_image_resized = cv2.resize(gray_image, target_size)

    # Apply a colormap to colorize the resized grayscale image
    color_image = cv2.applyColorMap(gray_image_resized,
cv2.COLORMAP_JET)

    # Save the colorized image to the output path (Output folder)
    cv2.imwrite(output_path, color_image)

    return color_image, gray_image_resized

# Function to process all images in the Gray folder
def process_images():
    # Determine a target size based on the first image in the Gray
folder
    first_image_path = os.path.join(gray_folder,
os.listdir(gray_folder)[0])
    first_gray_image = cv2.imread(first_image_path,
cv2.IMREAD_GRAYSCALE)
    target_size = (first_gray_image.shape[1],
first_gray_image.shape[0])  # (width, height)

    for filename in os.listdir(gray_folder):
        gray_image_path = os.path.join(gray_folder, filename)

        # Ensure the image file is a valid file
        if os.path.isfile(gray_image_path):
            # Define the output path for the colorized image in the
Output folder
            colorized_image_path = os.path.join(output_folder,
filename)
```

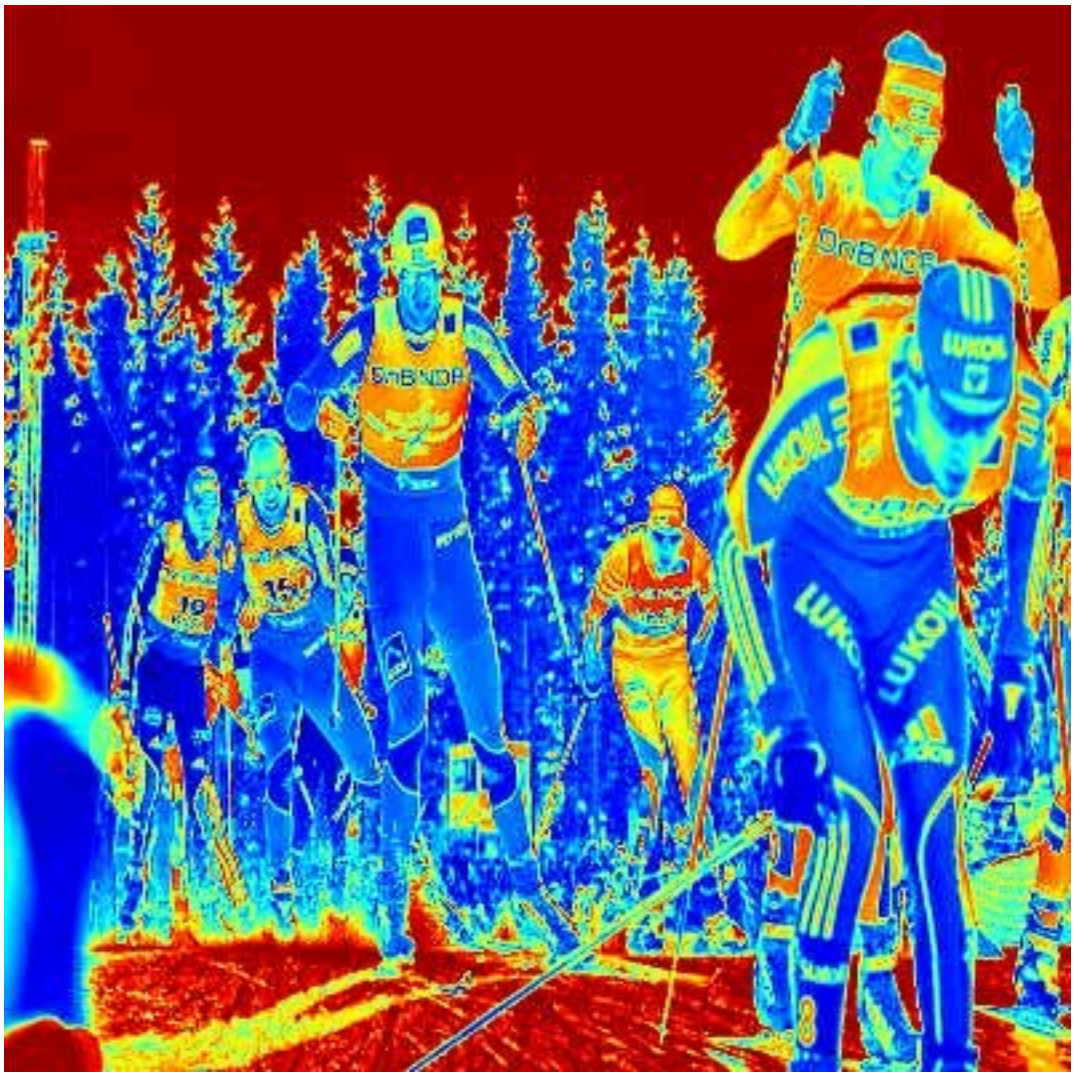
```
            # Colorize the image and get the colorized and resized
original grayscale image
            colorized_image, resized_gray_image =
colorize_image(gray_image_path, colorized_image_path, target_size)

            if colorized_image is not None:
                # Display the colorized image
                cv2_imshow(colorized_image)

                # Compute metrics
                mse, psnr, ssim_val =
compute_metrics(resized_gray_image, colorized_image)

                # Display metrics
                print(f"Metrics for {filename}:")
                print(f"MSE: {mse:.2f}")
                print(f"PSNR: {psnr:.2f} dB")
                print(f"SSIM: {ssim_val:.4f}")

# Save the model
save_model()

# Load the saved model
model = load_model(model_path)

# Call the function to process all images
process_images()
```

```
Model saved at /content/dataset/colorization_model.pth

<ipython-input-8-0fa258306fd8>:43: FutureWarning: You are using
`torch.load` with `weights_only=False` (the current default value),
which uses the default pickle module implicitly. It is possible to
construct malicious pickle data which will execute arbitrary code
during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  model.load_state_dict(torch.load(model_path,
map_location=torch.device('cpu')))
```

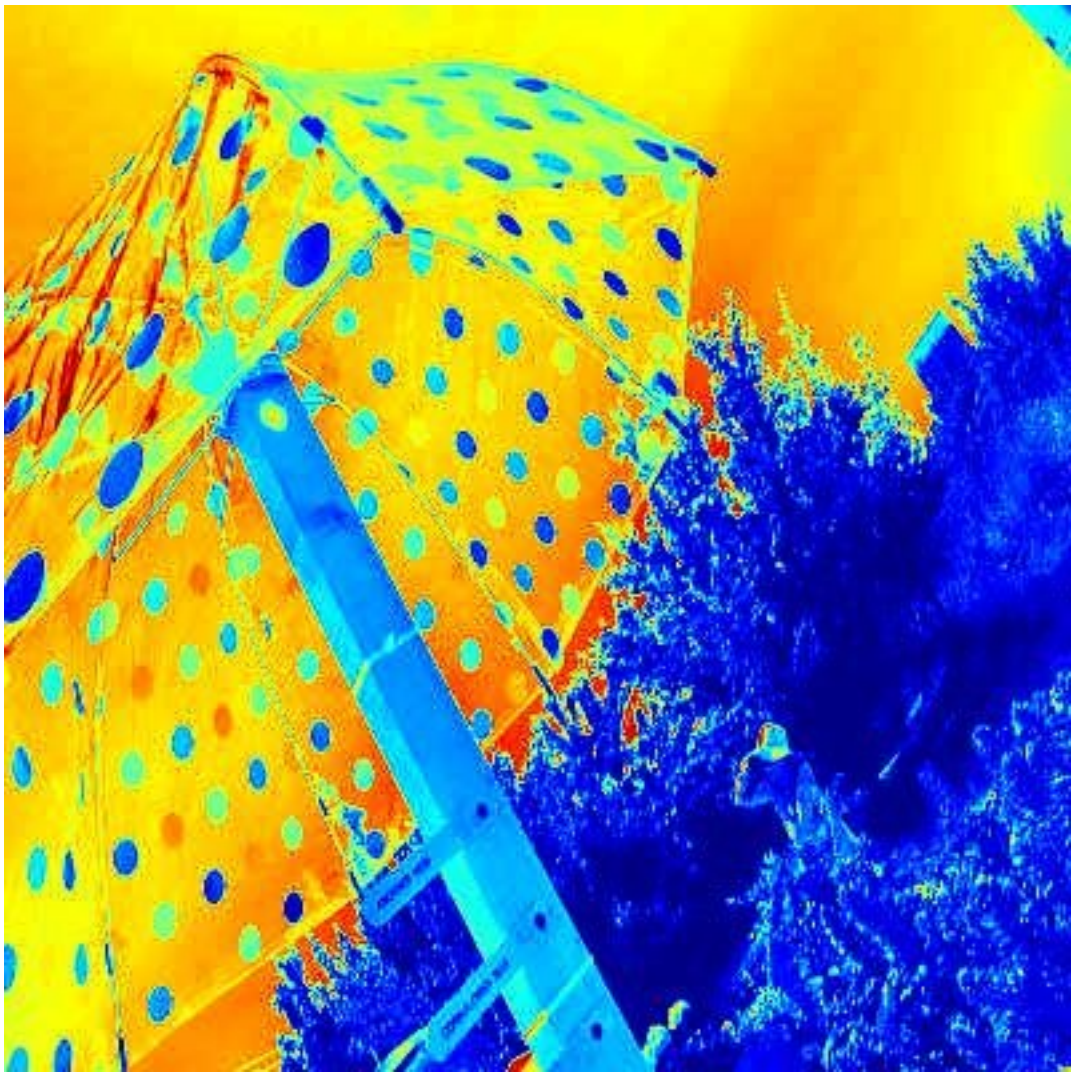

```
Metrics for image5018.jpg:
MSE: 117.90
PSNR: 8.83 dB
SSIM: 0.5057
```



```
Metrics for image5011.jpg:
MSE: 103.73
PSNR: 11.24 dB
SSIM: 0.4516
```



```
Metrics for image5014.jpg:
MSE: 87.02
PSNR: 7.00 dB
SSIM: 0.3463
```

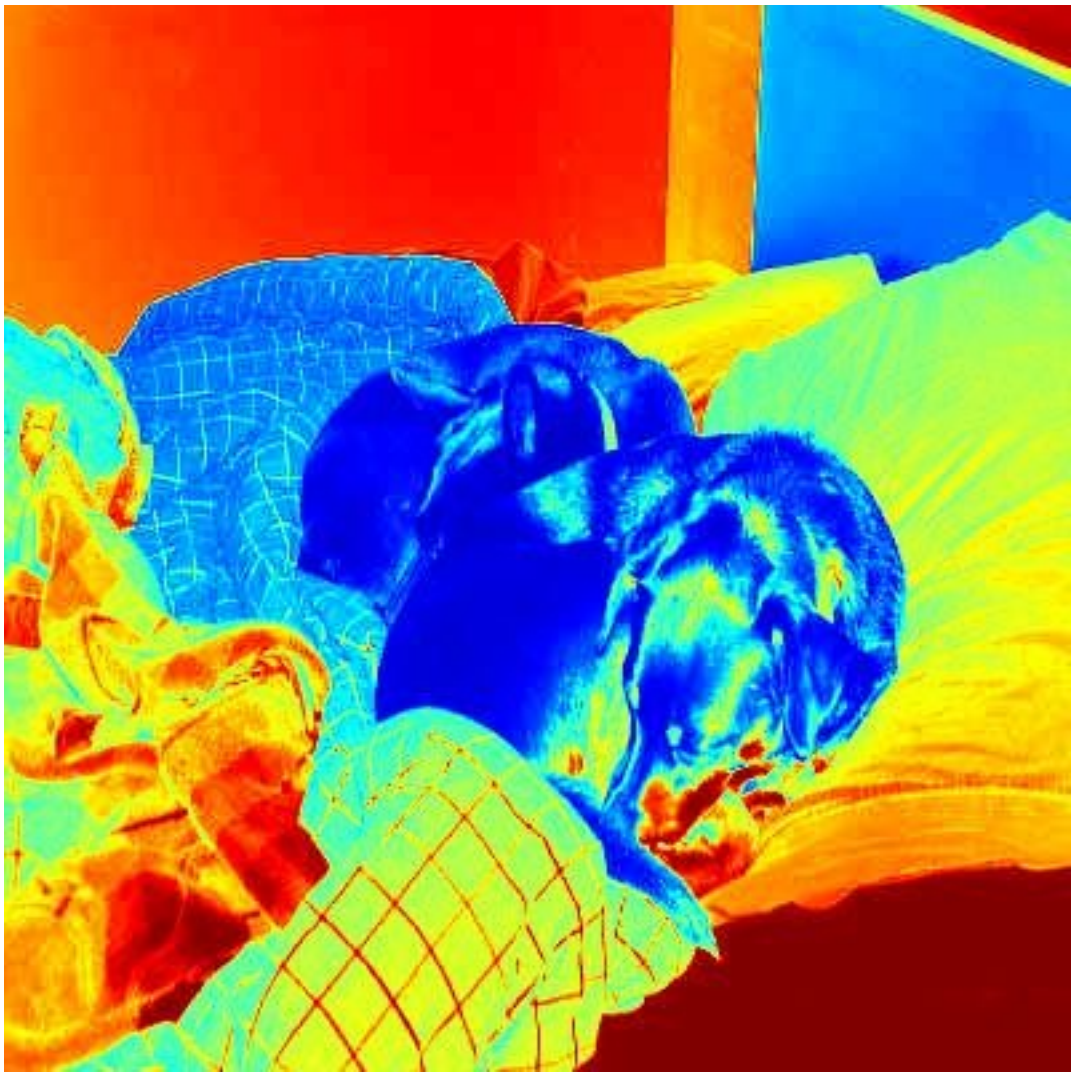


```
Metrics for image5012.jpg:
MSE: 99.41
PSNR: 15.09 dB
SSIM: 0.4557
```

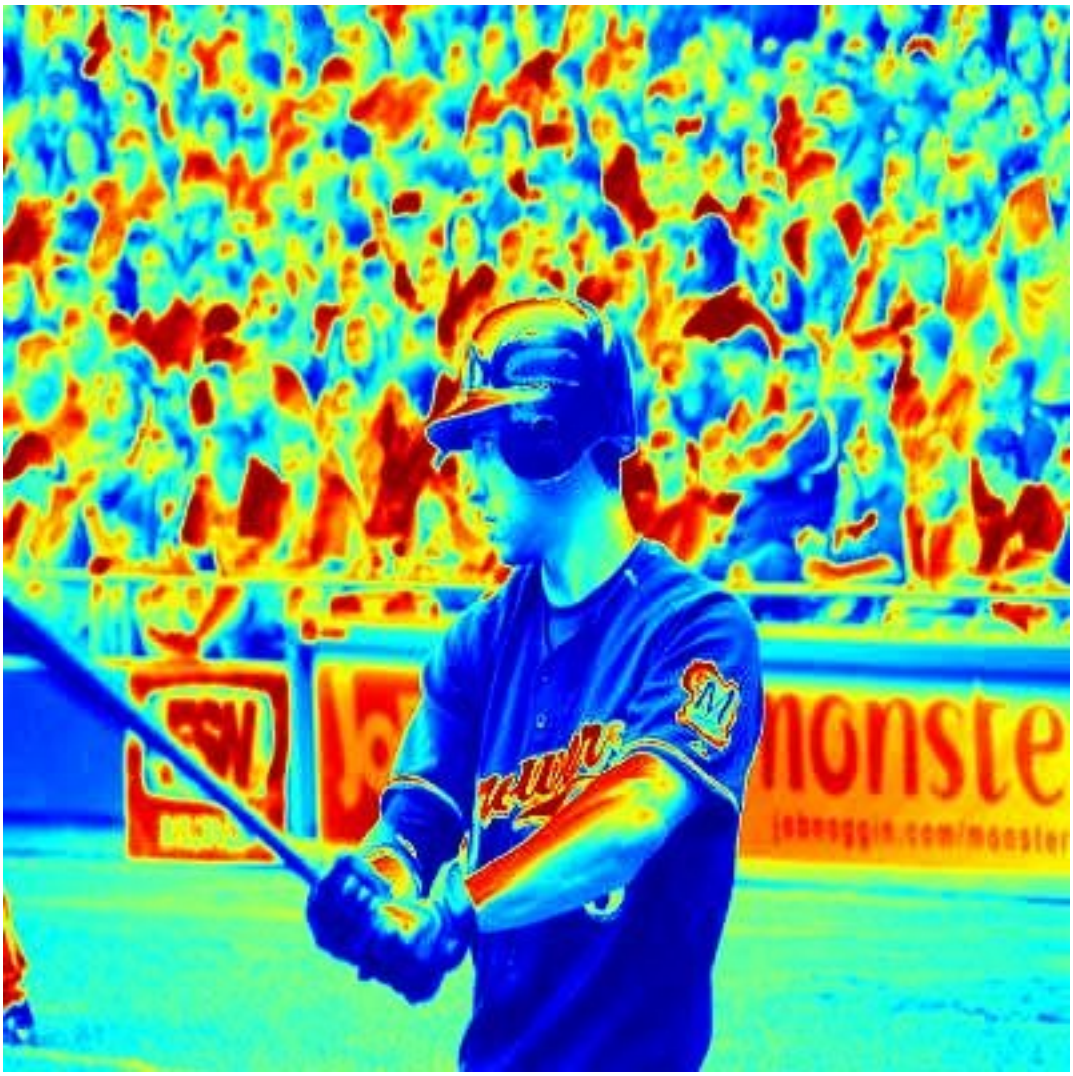
```
Metrics for image5009.jpg:
MSE: 109.51
PSNR: 8.40 dB
SSIM: 0.4523
```


```
Metrics for image5003.jpg:
MSE: 100.96
PSNR: 8.38 dB
SSIM: 0.5125
```

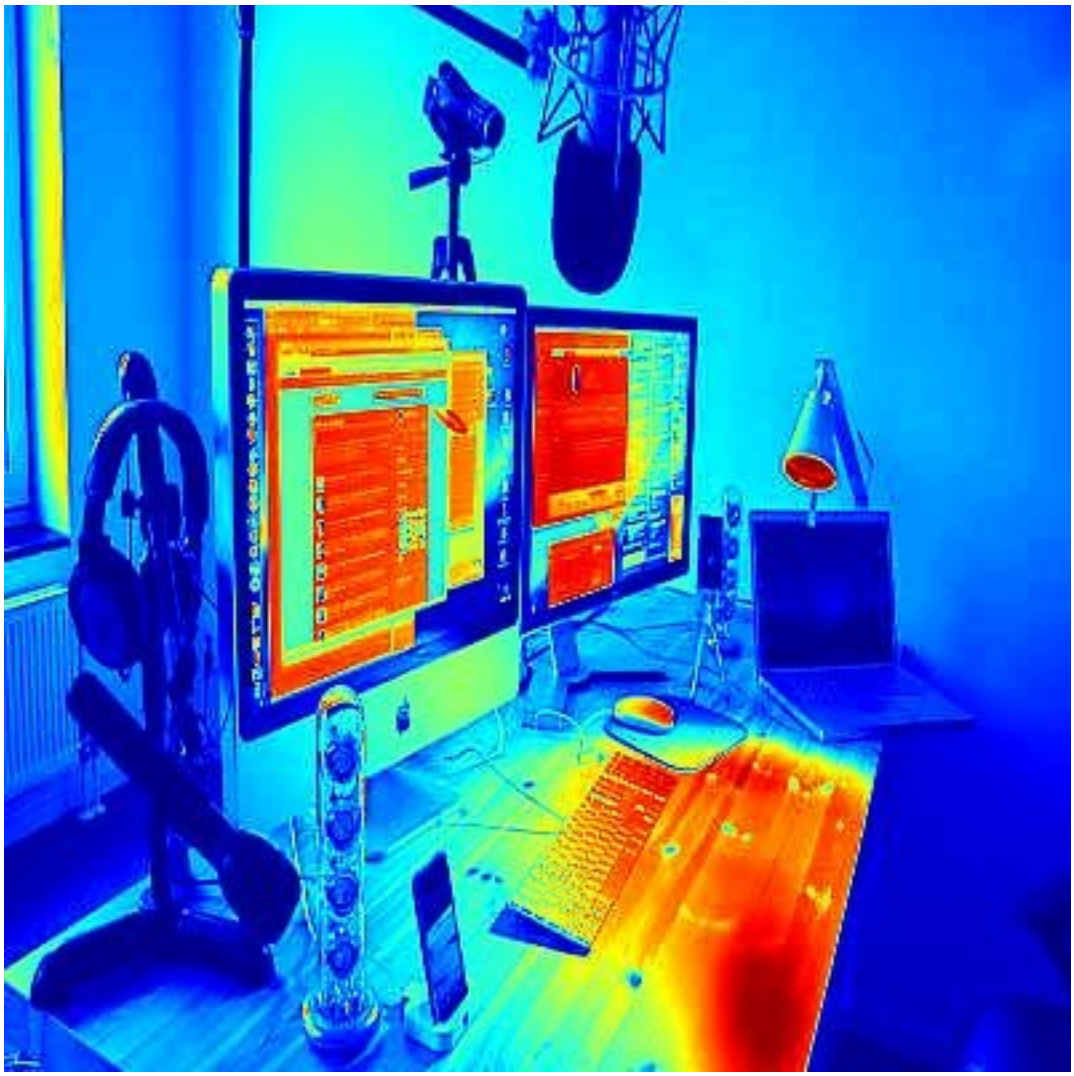
```
Metrics for image5021.jpg:
MSE: 121.35
PSNR: 7.51 dB
SSIM: 0.5014
```



```
Metrics for image5015.jpg:
MSE: 107.95
PSNR: 8.90 dB
SSIM: 0.3924
```

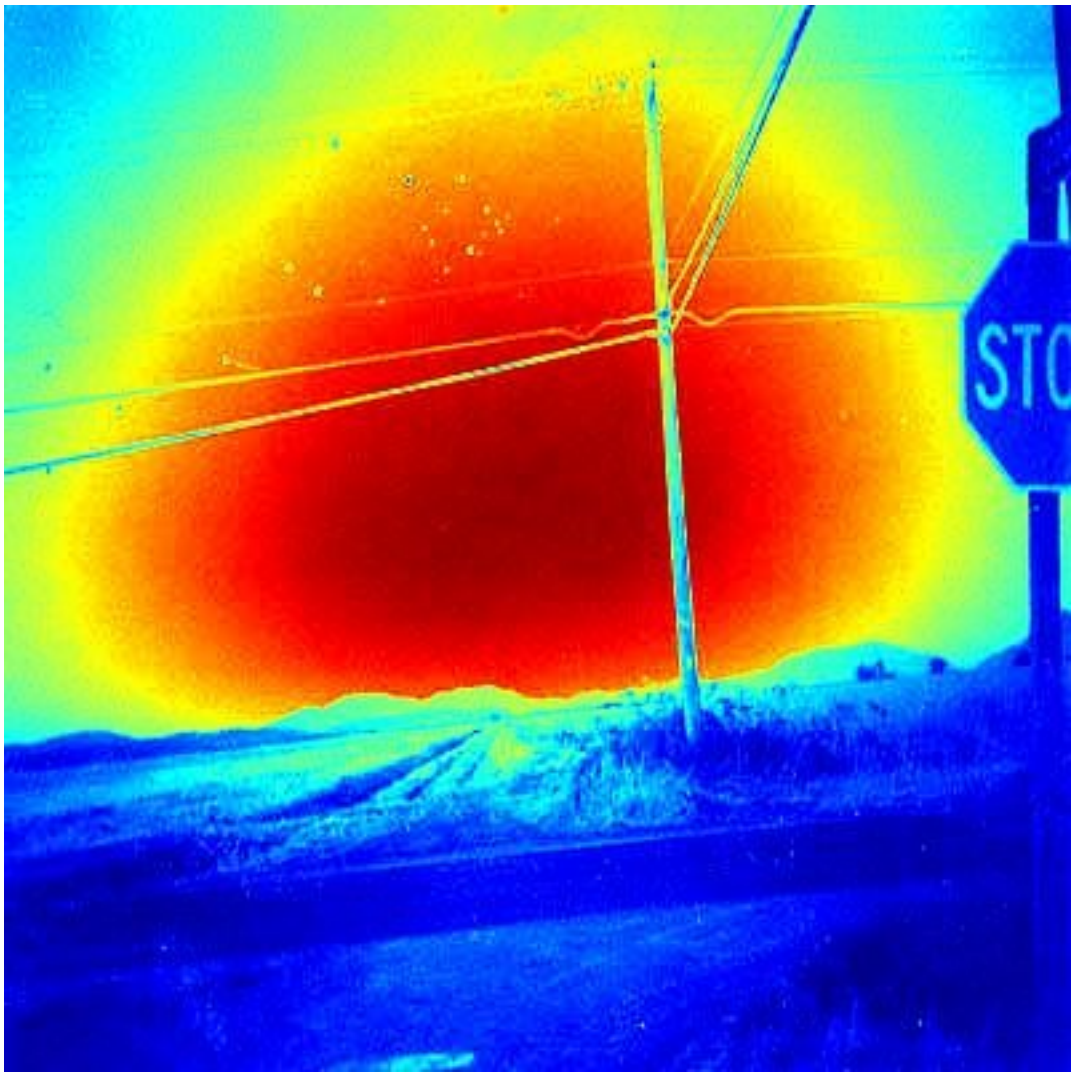


```
Metrics for image5007.jpg:
MSE: 106.12
PSNR: 10.60 dB
SSIM: 0.3169
```

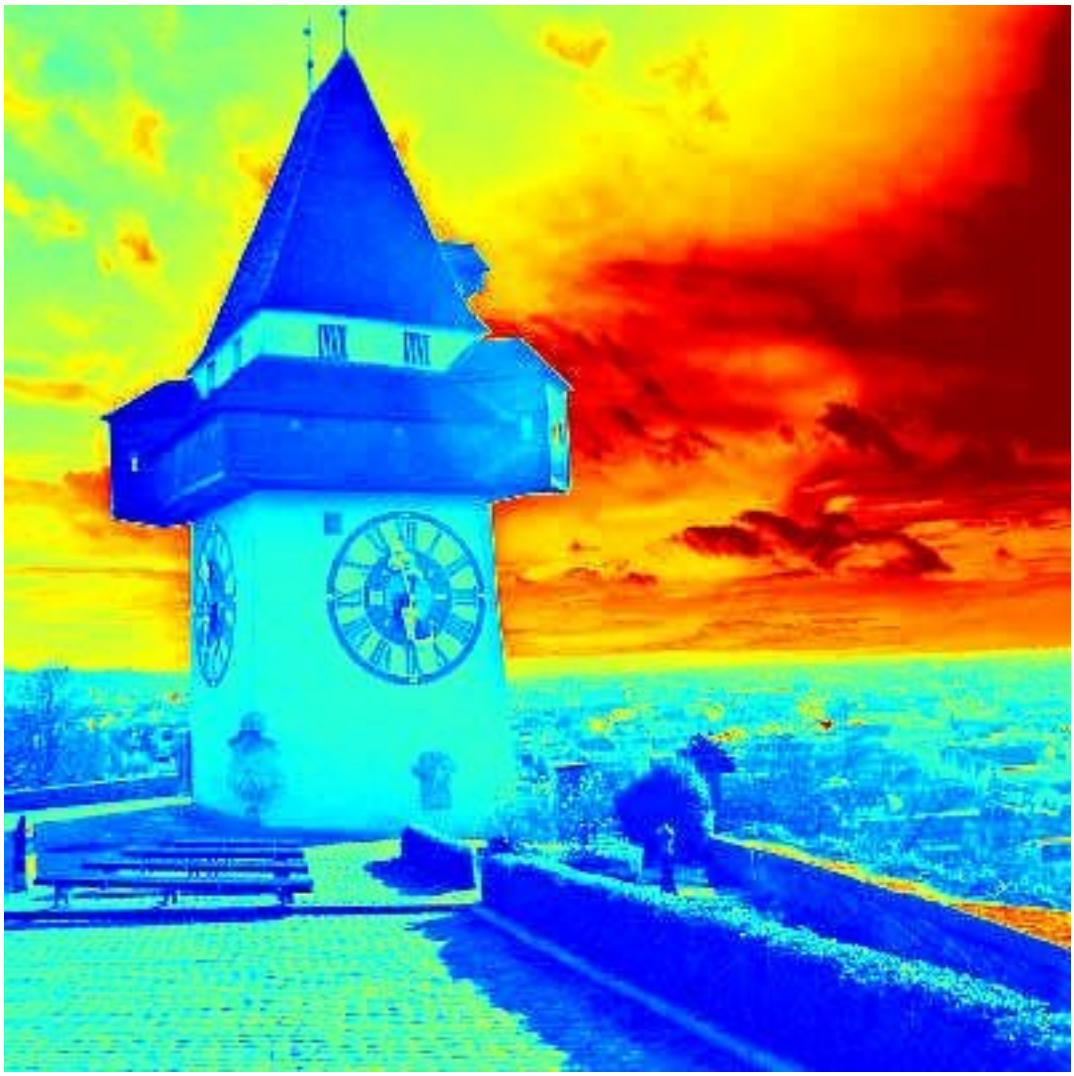


```
Metrics for image5006.jpg:
MSE: 102.06
PSNR: 12.83 dB
SSIM: 0.6121
```

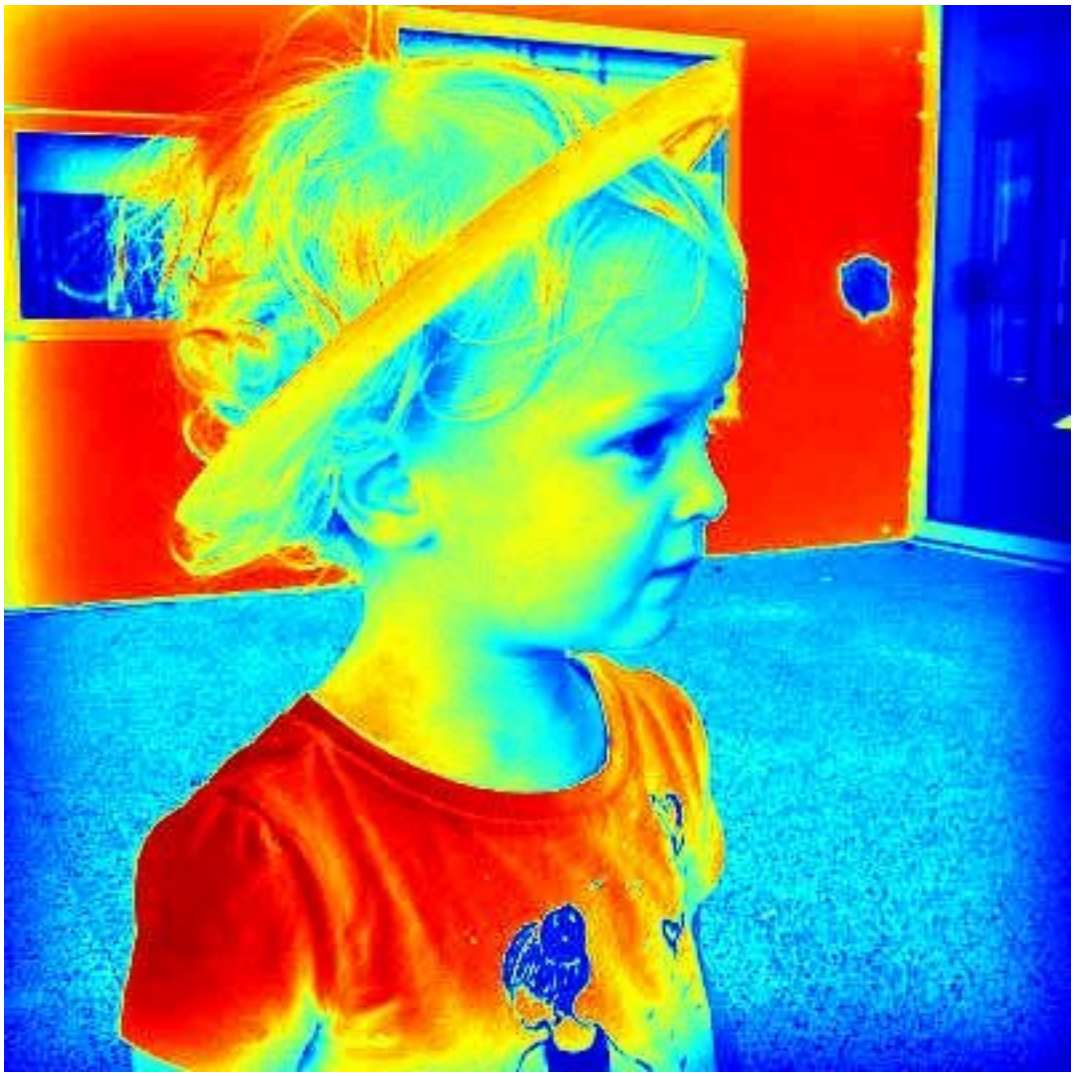
```
Metrics for image5010.jpg:
MSE: 91.65
PSNR: 11.43 dB
SSIM: 0.3261
```



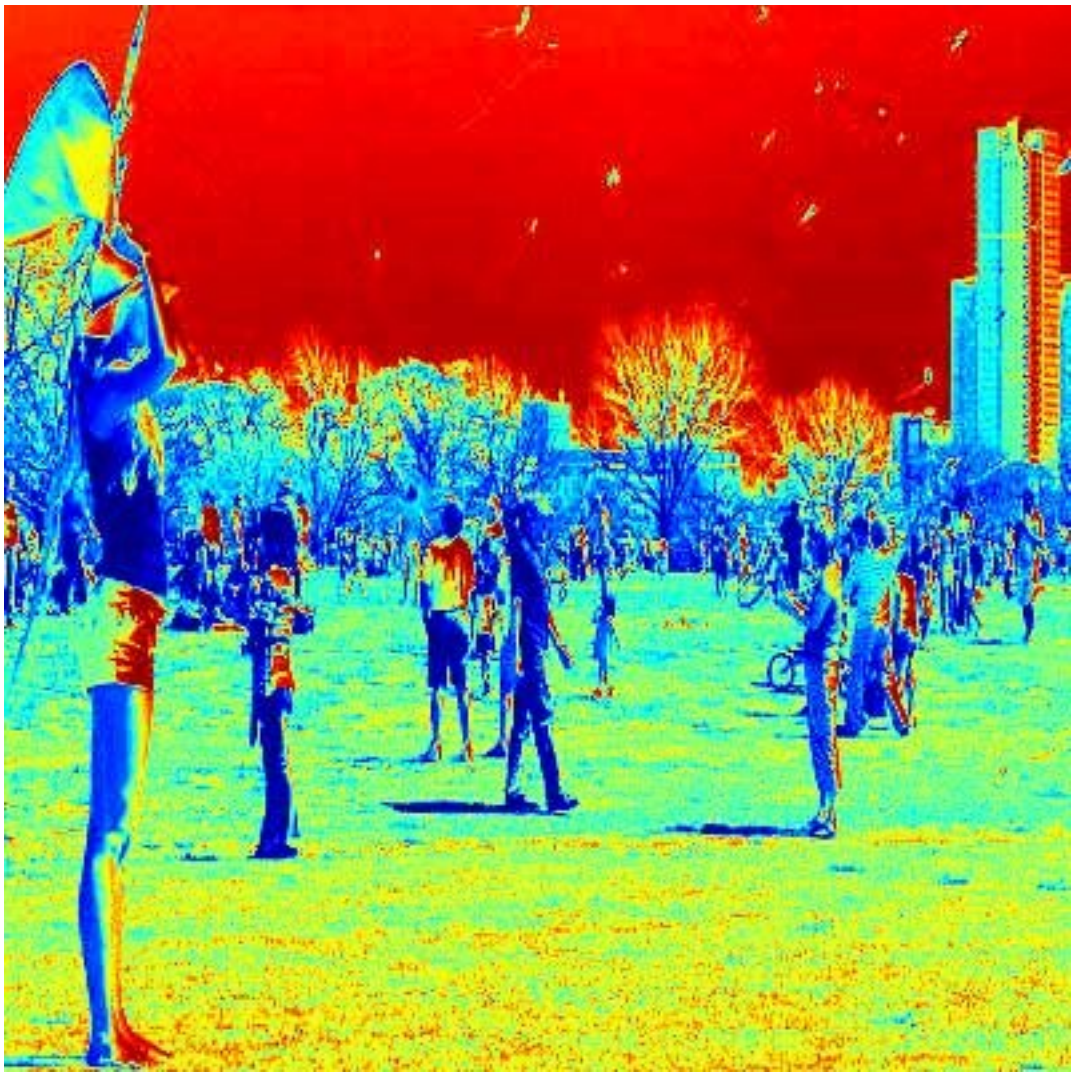

```
Metrics for image5000.jpg:
MSE: 93.53
PSNR: 9.77 dB
SSIM: 0.5159
```



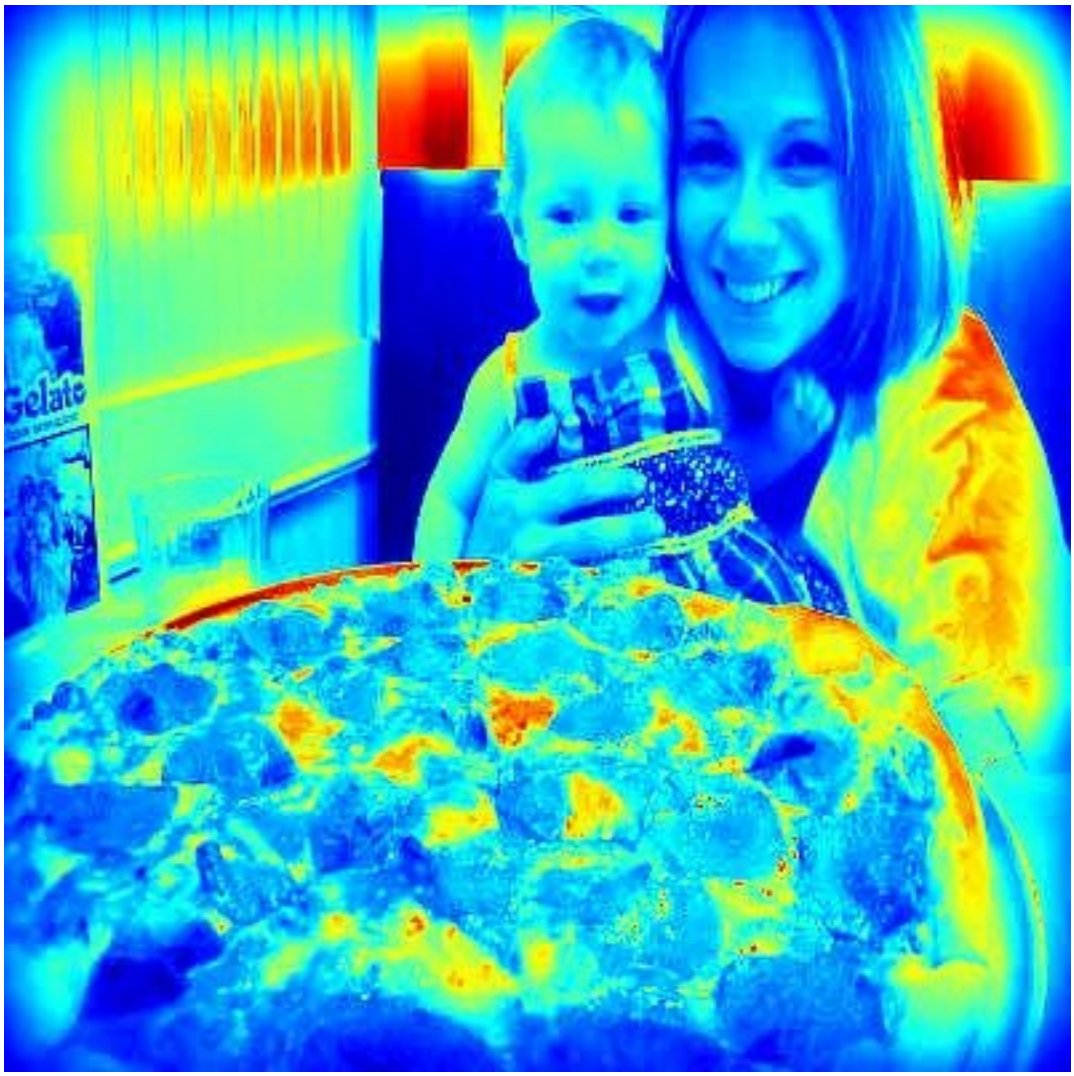
```
Metrics for image5013.jpg:
MSE: 107.89
PSNR: 9.40 dB
SSIM: 0.5356
```



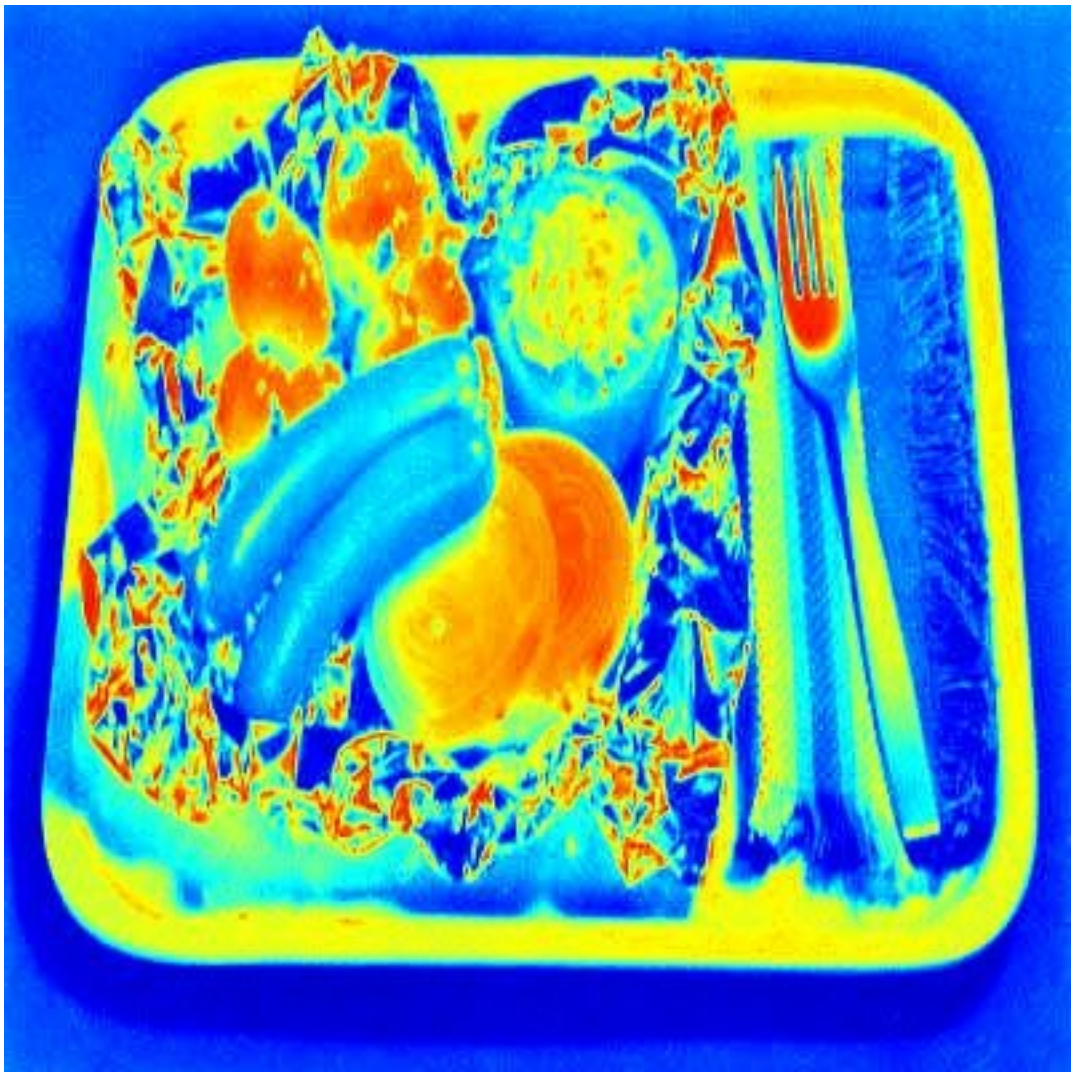
```
Metrics for image5019.jpg:
MSE: 101.07
PSNR: 9.83 dB
SSIM: 0.4345
```



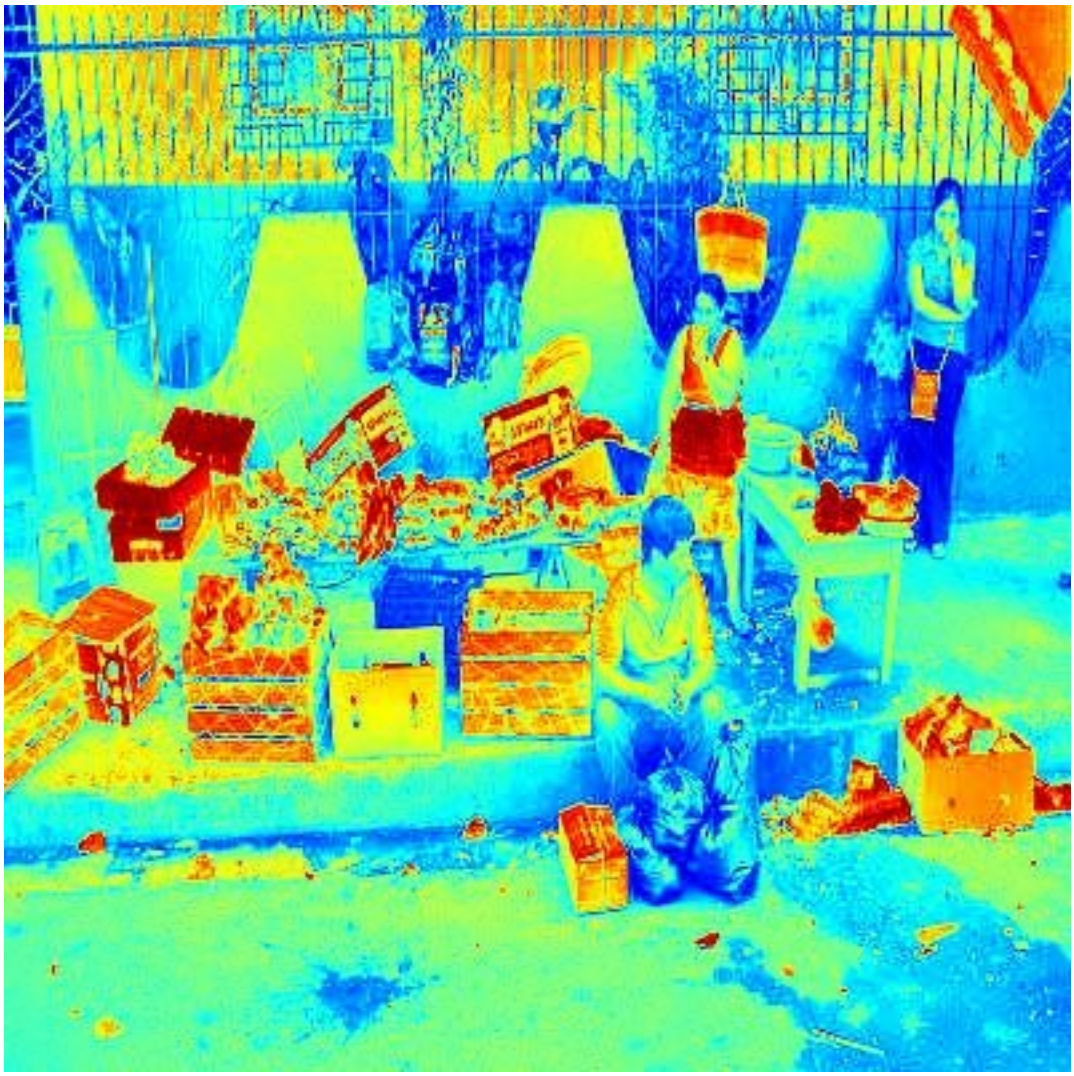
```
Metrics for image5020.jpg:
MSE: 108.68
PSNR: 7.72 dB
SSIM: 0.3642
```



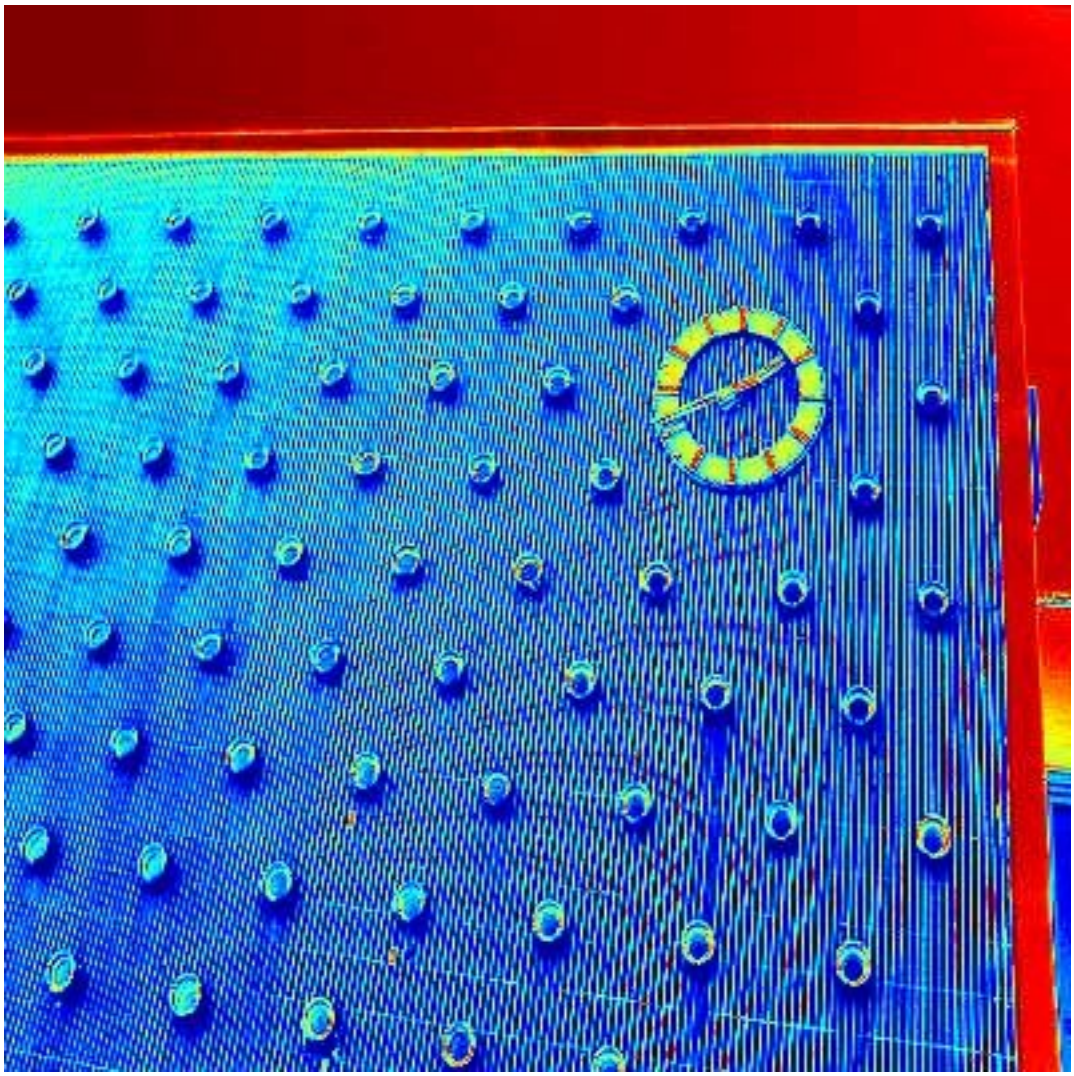

```
Metrics for image5005.jpg:
MSE: 108.01
PSNR: 12.17 dB
SSIM: 0.6348
```



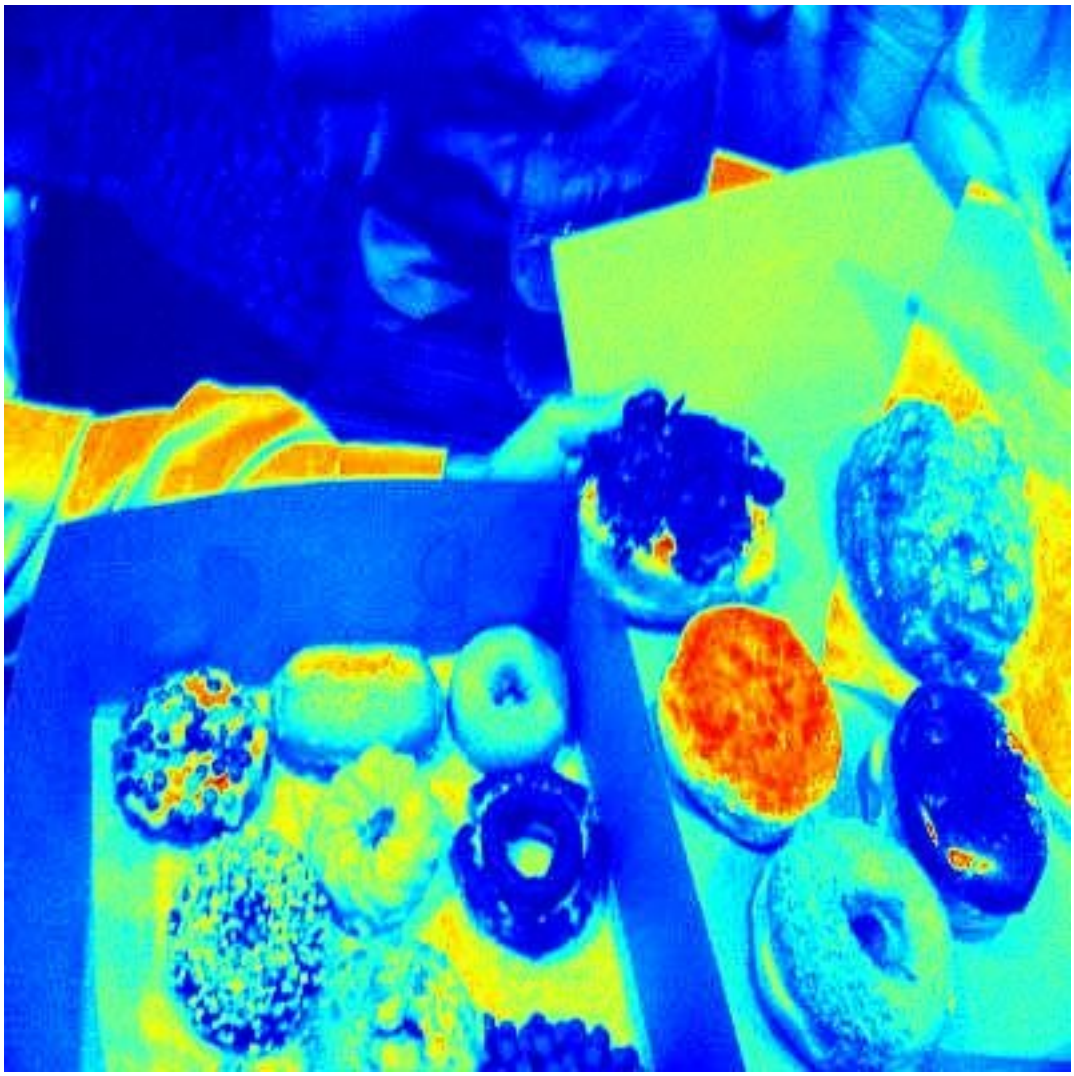
```
Metrics for image5001.jpg:
MSE: 97.00
PSNR: 13.70 dB
SSIM: 0.5944
```



```
Metrics for image5004.jpg:
MSE: 115.30
PSNR: 11.03 dB
SSIM: 0.4924
```



```
Metrics for image5017.jpg:
MSE: 100.12
PSNR: 9.06 dB
SSIM: 0.6437
```



```
Metrics for image5016.jpg:
MSE: 91.57
PSNR: 13.95 dB
SSIM: 0.7309
```

```
Metrics for image5002.jpg:
MSE: 90.83
PSNR: 13.36 dB
SSIM: 0.6294
```

```
Metrics for image5008.jpg:
MSE: 106.51
PSNR: 12.07 dB
SSIM: 0.6527
```

```
import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models
from sklearn.model_selection import train_test_split
from skimage.color import rgb2lab, lab2rgb
import matplotlib.pyplot as plt

# Paths
dataset_path = '/content/dataset/DataSET'
output_path = '/content/colorized_outputs'
os.makedirs(output_path, exist_ok=True)
```

```
# Load and preprocess dataset
def load_and_preprocess_images(dataset_path, image_size=(128, 128)):
    gray_images, color_images = [], []

    gray_folder = os.path.join(dataset_path, 'Gray')
    color_folder = os.path.join(dataset_path, 'Orignal')

    for file in os.listdir(gray_folder):
        if file.endswith(('.png', '.jpg', '.jpeg')):
            gray_path = os.path.join(gray_folder, file)
            color_path = os.path.join(color_folder, file)

            if os.path.exists(color_path):  # Ensure corresponding
color image exists
                gray_img = cv2.imread(gray_path, cv2.IMREAD_GRAYSCALE)
                color_img = cv2.imread(color_path)
                color_img = cv2.cvtColor(color_img, cv2.COLOR_BGR2RGB)

                # Resize images
                gray_img = cv2.resize(gray_img, image_size)
                color_img = cv2.resize(color_img, image_size)

                gray_images.append(gray_img)
                color_images.append(color_img)

    return np.array(gray_images), np.array(color_images)

# Normalize and prepare data
gray_images, color_images = load_and_preprocess_images(dataset_path)
gray_images = gray_images / 255.0  # Normalize grayscale images
color_images = color_images / 255.0  # Normalize color images

# Convert to LAB color space
lab_images = np.array([rgb2lab(img) for img in color_images])
L_images = lab_images[:, :, :, 0][..., np.newaxis]  # Extract L
channel and reshape
AB_images = lab_images[:, :, :, 1:] / 128.0  # Normalize AB channels
to [-1, 1]

# Split dataset
X_train, X_val, y_train, y_val = train_test_split(L_images, AB_images,
test_size=0.2, random_state=42)

# Build improved autoencoder
def build_autoencoder(input_shape):
    inputs = layers.Input(shape=input_shape)

    # Encoder
    x = layers.Conv2D(64, (3, 3), activation='relu', padding='same')
```

```
(inputs)
    x = layers.MaxPooling2D((2, 2), padding='same')(x)
    x = layers.Conv2D(128, (3, 3), activation='relu', padding='same')
(x)
    x = layers.MaxPooling2D((2, 2), padding='same')(x)
    x = layers.Conv2D(256, (3, 3), activation='relu', padding='same')
(x)
    encoded = layers.MaxPooling2D((2, 2), padding='same')(x)

    # Decoder
    x = layers.Conv2D(256, (3, 3), activation='relu', padding='same')
(encoded)
    x = layers.UpSampling2D((2, 2))(x)
    x = layers.Conv2D(128, (3, 3), activation='relu', padding='same')
(x)
    x = layers.UpSampling2D((2, 2))(x)
    x = layers.Conv2D(64, (3, 3), activation='relu', padding='same')
(x)
    x = layers.UpSampling2D((2, 2))(x)
    outputs = layers.Conv2D(2, (3, 3), activation='tanh',
padding='same')(x)  # Output AB channels

    return models.Model(inputs, outputs)

# Model configuration
input_shape = (128, 128, 1)
autoencoder = build_autoencoder(input_shape)
autoencoder.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1
e-4), loss=tf.keras.losses.MeanSquaredError())
autoencoder.summary()

# Train the model
history = autoencoder.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32
)

# Save the model
model_path = os.path.join(output_path, "colorization_autoencoder.h5")
autoencoder.save(model_path)
print(f"Model saved at {model_path}")

# Function to colorize images and display
def colorize_and_display(model, grayscale_images, output_folder,
n_images=5):
    os.makedirs(output_folder, exist_ok=True)

    for idx, gray_image in enumerate(grayscale_images[:n_images]):
```

```python
        gray_image_2d = gray_image.squeeze()  # Ensure the grayscale
image is 2D

        # Predict AB channels
        predicted_ab = model.predict(gray_image_2d[np.newaxis, :, :,
np.newaxis])[0]
        predicted_ab = predicted_ab * 128.0  # Denormalize AB channels

        # Combine L and predicted AB channels
        lab_image = np.concatenate([gray_image_2d[:, :, np.newaxis],
predicted_ab], axis=-1)
        colorized_image = lab2rgb(lab_image)

        # Plot original grayscale and colorized images
        plt.figure(figsize=(8, 4))

        # Grayscale image
        plt.subplot(1, 2, 1)
        plt.imshow(gray_image_2d, cmap='gray')
        plt.title("Grayscale Image")
        plt.axis("off")

        # Colorized image
        plt.subplot(1, 2, 2)
        plt.imshow(colorized_image)
        plt.title("Colorized by Model")
        plt.axis("off")

        plt.show()

# Test and display results
colorize_and_display(autoencoder, X_val, output_path)
```

```
Model: "functional"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳
━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                          ┃ Output Shape                ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇
━━━━━━━━━━━━━━━━━┩
│ input_layer (InputLayer)              │ (None, 128, 128, 1)         │
0 │
├───────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ conv2d (Conv2D)                       │ (None, 128, 128, 64)        │
640 │
├───────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ max_pooling2d (MaxPooling2D)          │ (None, 64, 64, 64)          │
```

```
0 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ conv2d_1 (Conv2D)                    │ (None, 64, 64, 128)         │
73,856 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ max_pooling2d_1 (MaxPooling2D)       │ (None, 32, 32, 128)         │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ conv2d_2 (Conv2D)                    │ (None, 32, 32, 256)         │
295,168 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ max_pooling2d_2 (MaxPooling2D)       │ (None, 16, 16, 256)         │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ conv2d_3 (Conv2D)                    │ (None, 16, 16, 256)         │
590,080 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ up_sampling2d (UpSampling2D)         │ (None, 32, 32, 256)         │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ conv2d_4 (Conv2D)                    │ (None, 32, 32, 128)         │
295,040 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ up_sampling2d_1 (UpSampling2D)       │ (None, 64, 64, 128)         │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ conv2d_5 (Conv2D)                    │ (None, 64, 64, 64)          │
73,792 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ up_sampling2d_2 (UpSampling2D)       │ (None, 128, 128, 64)        │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
│ conv2d_6 (Conv2D)                    │ (None, 128, 128, 2)         │
1,154 │
└──────────────────────────────────────┴─────────────────────────────┴
──────────────────┘

 Total params: 1,329,730 (5.07 MB)
```

```
 Trainable params: 1,329,730 (5.07 MB)

 Non-trainable params: 0 (0.00 B)

Epoch 1/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 11s 11s/step - loss: 0.6994 - val_loss:
0.7390
Epoch 2/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.8215 - val_loss: 0.7495
Epoch 3/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step - loss: 0.8351 - val_loss:
0.6580
Epoch 4/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - loss: 0.7134 - val_loss:
0.8855
Epoch 5/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - loss: 0.8104 - val_loss:
0.8930
Epoch 6/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 137ms/step - loss: 0.8115 - val_loss:
0.6567
Epoch 7/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 137ms/step - loss: 0.6184 - val_loss:
0.5701
Epoch 8/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step - loss: 0.6233 - val_loss:
0.4730
Epoch 9/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 88ms/step - loss: 0.5404 - val_loss:
0.5356
Epoch 10/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 153ms/step - loss: 0.5427 - val_loss:
0.4658
Epoch 11/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - loss: 0.4687 - val_loss:
0.1966
Epoch 12/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - loss: 0.2166 - val_loss:
0.1869
Epoch 13/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - loss: 0.1716 - val_loss:
0.3201
Epoch 14/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - loss: 0.2640 - val_loss:
0.2567
Epoch 15/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - loss: 0.1982 - val_loss:
0.1250
Epoch 16/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 147ms/step - loss: 0.0930 - val_loss:
```

```
0.1093
Epoch 17/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - loss: 0.1118 - val_loss:
0.1350
Epoch 18/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - loss: 0.1509 - val_loss:
0.1191
Epoch 19/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - loss: 0.1371 - val_loss:
0.0828
Epoch 20/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 128ms/step - loss: 0.0895 - val_loss:
0.0860
Epoch 21/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 152ms/step - loss: 0.0640 - val_loss:
0.1387
Epoch 22/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - loss: 0.0918 - val_loss:
0.1455
Epoch 23/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - loss: 0.0976 - val_loss:
0.1044
Epoch 24/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - loss: 0.0727 - val_loss:
0.0634
Epoch 25/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 149ms/step - loss: 0.0527 - val_loss:
0.0527
Epoch 26/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - loss: 0.0541 - val_loss:
0.0600
Epoch 27/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 148ms/step - loss: 0.0620 - val_loss:
0.0653
Epoch 28/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step - loss: 0.0628 - val_loss:
0.0594
Epoch 29/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - loss: 0.0516 - val_loss:
0.0526
Epoch 30/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - loss: 0.0402 - val_loss:
0.0564
Epoch 31/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - loss: 0.0407 - val_loss:
0.0650
Epoch 32/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step - loss: 0.0472 - val_loss:
0.0651
```

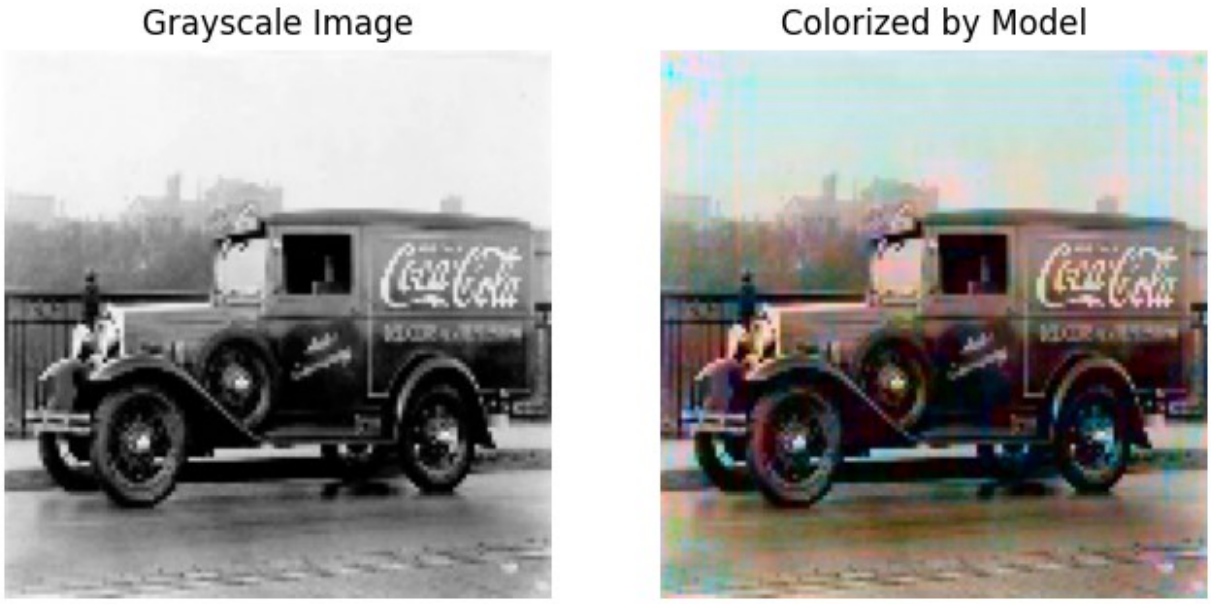
```
Epoch 33/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 136ms/step - loss: 0.0470 - val_loss:
0.0563
Epoch 34/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 142ms/step - loss: 0.0396 - val_loss:
0.0477
Epoch 35/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 145ms/step - loss: 0.0336 - val_loss:
0.0444
Epoch 36/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step - loss: 0.0335 - val_loss:
0.0436
Epoch 37/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 151ms/step - loss: 0.0360 - val_loss:
0.0413
Epoch 38/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - loss: 0.0364 - val_loss:
0.0380
Epoch 39/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 136ms/step - loss: 0.0341 - val_loss:
0.0361
Epoch 40/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - loss: 0.0310 - val_loss:
0.0376
Epoch 41/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 150ms/step - loss: 0.0289 - val_loss:
0.0425
Epoch 42/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - loss: 0.0290 - val_loss:
0.0474
Epoch 43/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - loss: 0.0303 - val_loss:
0.0481
Epoch 44/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 126ms/step - loss: 0.0302 - val_loss:
0.0444
Epoch 45/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - loss: 0.0284 - val_loss:
0.0393
Epoch 46/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - loss: 0.0266 - val_loss:
0.0355
Epoch 47/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - loss: 0.0259 - val_loss:
0.0339
Epoch 48/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step - loss: 0.0260 - val_loss:
0.0337
Epoch 49/100
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 139ms/step - loss: 0.0262 - val_loss:
0.0342
Epoch 50/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step - loss: 0.0261 - val_loss:
0.0346
Epoch 51/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step - loss: 0.0256 - val_loss:
0.0347
Epoch 52/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - loss: 0.0248 - val_loss:
0.0346
Epoch 53/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - loss: 0.0242 - val_loss:
0.0348
Epoch 54/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - loss: 0.0239 - val_loss:
0.0354
Epoch 55/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 135ms/step - loss: 0.0238 - val_loss:
0.0360
Epoch 56/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - loss: 0.0238 - val_loss:
0.0365
Epoch 57/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - loss: 0.0236 - val_loss:
0.0364
Epoch 58/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - loss: 0.0233 - val_loss:
0.0357
Epoch 59/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 150ms/step - loss: 0.0229 - val_loss:
0.0346
Epoch 60/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 141ms/step - loss: 0.0226 - val_loss:
0.0333
Epoch 61/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - loss: 0.0223 - val_loss:
0.0322
Epoch 62/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 126ms/step - loss: 0.0222 - val_loss:
0.0315
Epoch 63/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step - loss: 0.0222 - val_loss:
0.0313
Epoch 64/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - loss: 0.0220 - val_loss:
0.0314
Epoch 65/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step - loss: 0.0218 - val_loss:
```

```
0.0318
Epoch 66/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - loss: 0.0216 - val_loss:
0.0322
Epoch 67/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - loss: 0.0213 - val_loss:
0.0326
Epoch 68/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - loss: 0.0212 - val_loss:
0.0328
Epoch 69/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 104ms/step - loss: 0.0210 - val_loss:
0.0329
Epoch 70/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 139ms/step - loss: 0.0209 - val_loss:
0.0329
Epoch 71/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 135ms/step - loss: 0.0208 - val_loss:
0.0327
Epoch 72/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - loss: 0.0206 - val_loss:
0.0324
Epoch 73/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - loss: 0.0205 - val_loss:
0.0320
Epoch 74/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - loss: 0.0203 - val_loss:
0.0315
Epoch 75/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - loss: 0.0202 - val_loss:
0.0309
Epoch 76/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 131ms/step - loss: 0.0201 - val_loss:
0.0305
Epoch 77/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - loss: 0.0199 - val_loss:
0.0303
Epoch 78/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - loss: 0.0198 - val_loss:
0.0303
Epoch 79/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - loss: 0.0197 - val_loss:
0.0305
Epoch 80/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - loss: 0.0195 - val_loss:
0.0307
Epoch 81/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 127ms/step - loss: 0.0194 - val_loss:
0.0308
```

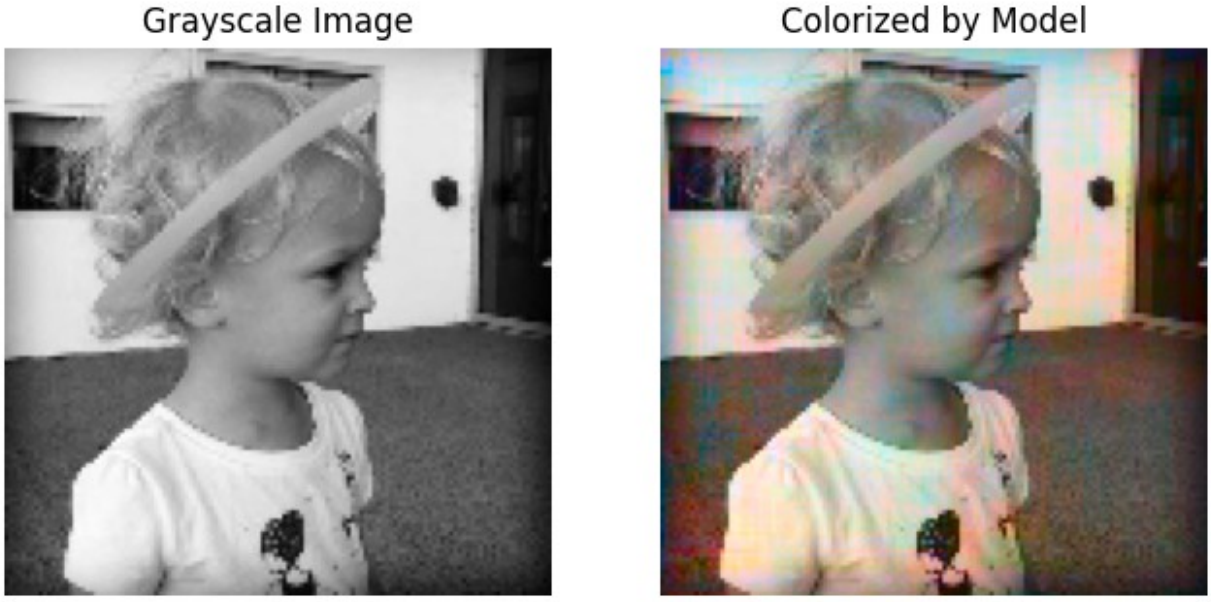
```
Epoch 82/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step - loss: 0.0193 - val_loss:
0.0307
Epoch 83/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - loss: 0.0192 - val_loss:
0.0305
Epoch 84/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - loss: 0.0191 - val_loss:
0.0302
Epoch 85/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step - loss: 0.0189 - val_loss:
0.0300
Epoch 86/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 306ms/step - loss: 0.0188 - val_loss:
0.0299
Epoch 87/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 298ms/step - loss: 0.0187 - val_loss:
0.0299
Epoch 88/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 285ms/step - loss: 0.0186 - val_loss:
0.0298
Epoch 89/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 128ms/step - loss: 0.0185 - val_loss:
0.0297
Epoch 90/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step - loss: 0.0184 - val_loss:
0.0297
Epoch 91/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - loss: 0.0183 - val_loss:
0.0298
Epoch 92/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 137ms/step - loss: 0.0181 - val_loss:
0.0297
Epoch 93/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - loss: 0.0180 - val_loss:
0.0294
Epoch 94/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - loss: 0.0179 - val_loss:
0.0292
Epoch 95/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - loss: 0.0178 - val_loss:
0.0291
Epoch 96/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 144ms/step - loss: 0.0177 - val_loss:
0.0291
Epoch 97/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - loss: 0.0176 - val_loss:
0.0289
Epoch 98/100
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step - loss: 0.0175 - val_loss:
0.0287
Epoch 99/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - loss: 0.0174 - val_loss:
0.0286
Epoch 100/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - loss: 0.0173 - val_loss:
0.0285

WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.

You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered
legacy. We recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.
Model saved at /content/colorized_outputs/colorization_autoencoder.h5
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step

<ipython-input-9-133dc37c4f7a>:109: UserWarning: Conversion from CIE-
LAB, via XYZ to sRGB color space resulted in 12 negative Z values that
have been clipped to zero
  colorized_image = lab2rgb(lab_image)
```



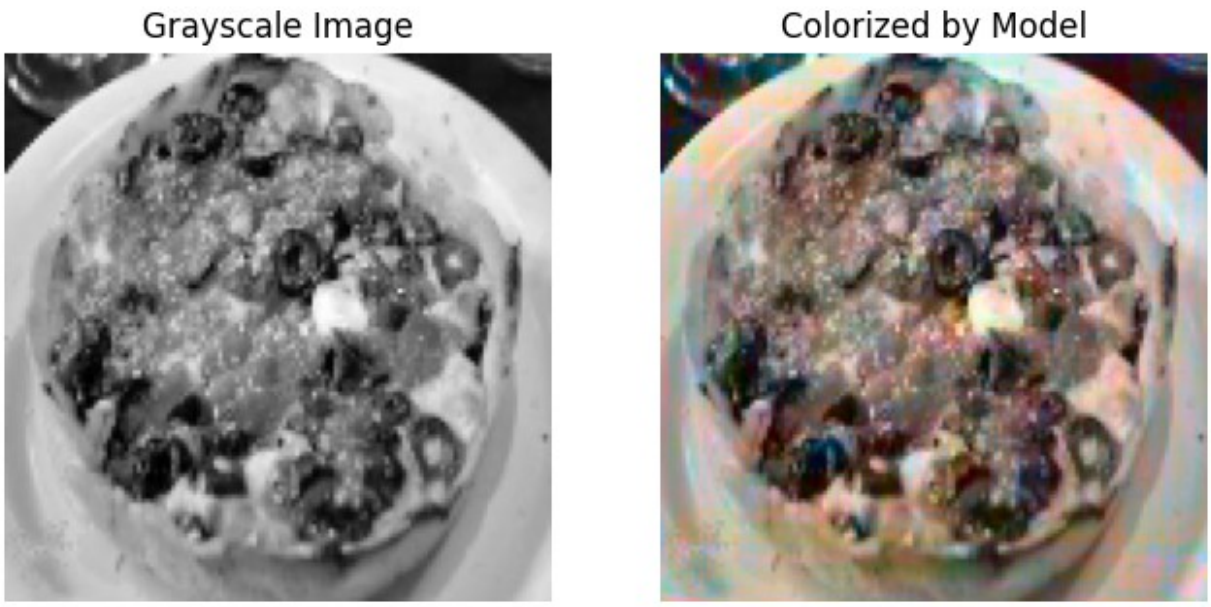

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
```

```
<ipython-input-9-133dc37c4f7a>:109: UserWarning: Conversion from CIE-LAB, via XYZ to sRGB color space resulted in 1 negative Z values that have been clipped to zero
  colorized_image = lab2rgb(lab_image)
```




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 150ms/step
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
```




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step

<ipython-input-9-133dc37c4f7a>:109: UserWarning: Conversion from CIE-LAB, via XYZ to sRGB color space resulted in 2 negative Z values that have been clipped to zero
  colorized_image = lab2rgb(lab_image)
```


```
from pyngrok import ngrok, conf
import os

# Set the ngrok auth token
```

```
conf.get_default().auth_token =
"2phP00Avhicmtseo8jiNS52EH7C_7zSVMRrbX8hTNNWcjpjet"

# Run the Streamlit app
os.system(r"streamlit run C:\Users\altaf\Desktop\Colorisation\
colorization_env\dashboard.py &")

# Connect ngrok to the correct port
public_url = ngrok.connect(8501)  # 8501 is the default Streamlit port
print("Public URL:", public_url)
```

```
INFO:pyngrok.ngrok:Opening tunnel named: http-8501-04766ad8-34ce-4011-
b016-d958fbbe5e54
```

```
Opening tunnel named: http-8501-04766ad8-34ce-4011-b016-d958fbbe5e54
```

```
INFO:pyngrok.process:Overriding default auth token
```

```
Overriding default auth token
```

```
INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:33+0000 lvl=info msg="no
configuration paths supplied"
```

```
t=2024-12-03T20:22:33+0000 lvl=info msg="no configuration paths
supplied"
```

```
INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:33+0000 lvl=info
msg="using configuration at default config path"
path=/root/.config/ngrok/ngrok.yml
```

```
t=2024-12-03T20:22:33+0000 lvl=info msg="using configuration at
default config path" path=/root/.config/ngrok/ngrok.yml
```

```
INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:33+0000 lvl=info
msg="open config file" path=/root/.config/ngrok/ngrok.yml err=nil
```

```
t=2024-12-03T20:22:33+0000 lvl=info msg="open config file"
path=/root/.config/ngrok/ngrok.yml err=nil
```

```
INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:33+0000 lvl=info
msg="starting web service" obj=web addr=127.0.0.1:4040 allow_hosts=[]
```

```
t=2024-12-03T20:22:33+0000 lvl=info msg="starting web service" obj=web
addr=127.0.0.1:4040 allow_hosts=[]
```

```
INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info
msg="client session established" obj=tunnels.session
```

```
t=2024-12-03T20:22:34+0000 lvl=info msg="client session established"
obj=tunnels.session
```

```
INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info
msg="tunnel session started" obj=tunnels.session
```

```
t=2024-12-03T20:22:34+0000 lvl=info msg="tunnel session started"
obj=tunnels.session

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info
msg=start pg=/api/tunnels id=5716a3ea887f9a44

t=2024-12-03T20:22:34+0000 lvl=info msg=start pg=/api/tunnels
id=5716a3ea887f9a44

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info msg=end
pg=/api/tunnels id=5716a3ea887f9a44 status=200 dur=454.33µs

t=2024-12-03T20:22:34+0000 lvl=info msg=end pg=/api/tunnels
id=5716a3ea887f9a44 status=200 dur=454.33µs

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info
msg=start pg=/api/tunnels id=4c7127b3dfaddae3

t=2024-12-03T20:22:34+0000 lvl=info msg=start pg=/api/tunnels
id=4c7127b3dfaddae3

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info msg=end
pg=/api/tunnels id=4c7127b3dfaddae3 status=200 dur=133.506µs

t=2024-12-03T20:22:34+0000 lvl=info msg=end pg=/api/tunnels
id=4c7127b3dfaddae3 status=200 dur=133.506µs

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info
msg=start pg=/api/tunnels id=01c92830571a2ec9

t=2024-12-03T20:22:34+0000 lvl=info msg=start pg=/api/tunnels
id=01c92830571a2ec9
Public URL: NgrokTunnel: "https://7c97-34-83-23-10.ngrok-free.app" ->
"http://localhost:8501"

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info
msg="started tunnel" obj=tunnels name=http-8501-04766ad8-34ce-4011-
b016-d958fbbe5e54 addr=http://localhost:8501 url=https://7c97-34-83-
23-10.ngrok-free.app

t=2024-12-03T20:22:34+0000 lvl=info msg="started tunnel" obj=tunnels
name=http-8501-04766ad8-34ce-4011-b016-d958fbbe5e54
addr=http://localhost:8501 url=https://7c97-34-83-23-10.ngrok-free.app

INFO:pyngrok.process.ngrok:t=2024-12-03T20:22:34+0000 lvl=info msg=end
pg=/api/tunnels id=01c92830571a2ec9 status=201 dur=74.9065ms

t=2024-12-03T20:22:34+0000 lvl=info msg=end pg=/api/tunnels
id=01c92830571a2ec9 status=201 dur=74.9065ms

import gradio as gr
import torch
import tensorflow as tf
```

```
from torchvision.transforms import ToTensor, ToPILImage
from PIL import Image
import numpy as np

import torch.nn as nn

class ColorizationModelClass(nn.Module):
    def __init__(self):
        super(ColorizationModelClass, self).__init__()
        # Define the correct layers
        self.conv1 = nn.Conv2d(1, 64, kernel_size=3, stride=1,
padding=1)
        self.relu = nn.ReLU()
        self.conv2 = nn.Conv2d(64, 128, kernel_size=3, stride=1,
padding=1)
        self.conv3 = nn.Conv2d(128, 3, kernel_size=3, stride=1,
padding=1)  # Add the third convolutional layer

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.relu(self.conv2(x))
        x = self.conv3(x)
        return x

# Now instantiate the model
model1 = ColorizationModelClass()

# Load model weights into the updated architecture with strict=False
model1.load_state_dict(torch.load("/content/dataset/colorization_model
.pth", map_location=torch.device('cpu')), strict=False)
model1.eval()
```

```
<ipython-input-18-f0b1099e47b5>:2: FutureWarning: You are using
`torch.load` with `weights_only=False` (the current default value),
which uses the default pickle module implicitly. It is possible to
construct malicious pickle data which will execute arbitrary code
during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
```

```
model1.load_state_dict(torch.load("/content/dataset/colorization_model
.pth", map_location=torch.device('cpu')), strict=False)
```

```
ColorizationModelClass(
  (conv1): Conv2d(1, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
  (relu): ReLU()
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
  (conv3): Conv2d(128, 3, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
)
```

```
from PIL import Image
import numpy as np
import torch
import os

# Load grayscale image
image_path = "/content/dataset/DataSET/Gray/image5000.jpg"  # Replace
with your image path
try:
    image = Image.open(image_path).convert("L")
except FileNotFoundError:
    raise FileNotFoundError(f"Input image file not found at
{image_path}. Please provide the correct path.")

gray_image = np.array(image)

# Load Model 1
model1_path = "/content/dataset/colorization_model.pth"  # Replace
with your model path

class ColorizationModelClass(torch.nn.Module):
    def __init__(self):
        super(ColorizationModelClass, self).__init__()
        self.conv1 = torch.nn.Conv2d(1, 64, kernel_size=3, stride=1,
padding=1)
        self.conv2 = torch.nn.Conv2d(64, 128, kernel_size=3, stride=1,
padding=1)
        self.conv3 = torch.nn.Conv2d(128, 3, kernel_size=3, stride=1,
padding=1)

    def forward(self, x):
        x = torch.relu(self.conv1(x))
        x = torch.relu(self.conv2(x))
        x = torch.sigmoid(self.conv3(x))
        return x

# Initialize and load the model
model1 = ColorizationModelClass()

try:
```

```
    # Load model weights with strict=False to ignore missing or
mismatched layers
    model1.load_state_dict(torch.load(model1_path,
map_location=torch.device("cpu")), strict=False)
except FileNotFoundError:
    raise FileNotFoundError(f"Model file not found at {model1_path}.
Please provide the correct path.")

model1.eval()

# Preprocess and infer
input_tensor = torch.tensor(gray_image[np.newaxis, np.newaxis, :, :],
dtype=torch.float32) / 255.0
with torch.no_grad():  # Disable gradient calculations for inference
    output_tensor = model1(input_tensor).numpy()

# Convert output to image format
colorized_image = np.clip(output_tensor[0].transpose(1, 2, 0) * 255.0,
0, 255).astype(np.uint8)

# Save the result
output_dir = "/content/output"  # Set the directory to save the output
os.makedirs(output_dir, exist_ok=True)  # Create the directory if it
doesn't exist
output_path = os.path.join(output_dir, "colorized_output_model1.png")

try:
    Image.fromarray(colorized_image).save(output_path)
    print(f"Colorization completed successfully. Image saved at
{output_path}")
except Exception as e:
    raise IOError(f"Error saving image: {e}")
```

```
<ipython-input-20-4ee17826b552>:36: FutureWarning: You are using
`torch.load` with `weights_only=False` (the current default value),
which uses the default pickle module implicitly. It is possible to
construct malicious pickle data which will execute arbitrary code
during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  model1.load_state_dict(torch.load(model1_path,
map_location=torch.device("cpu")), strict=False)
```

```
Colorization completed successfully. Image saved at
/content/output/colorized_output_model1.png
```

```
import os
from PIL import Image
from deoldify.visualize import get_image_colorizer

# Define paths
image_path = "/content/dataset/DataSET/Gray/image5000.jpg"  # Replace
with your image path
input_path = "/tmp/input_image.jpg"
output_path = "/mnt/data/colorized_output_deoldify.png"  # Output path
for the colorized image

# Verify the image file exists
if not os.path.exists(image_path):
    print(f"Image file not found at {image_path}. Creating a dummy
grayscale image for testing.")
    # Create a dummy grayscale image if the file does not exist
    dummy_image = Image.new("L", (256, 256), 128)  # Medium brightness
gray
    dummy_image.save(image_path)
    print(f"Dummy image created at {image_path}.")

# Load the grayscale image
try:
    image = Image.open(image_path).convert("L")
    image.save(input_path)  # Save the input image for processing
    print(f"Input image saved at {input_path}.")
except Exception as e:
    raise IOError(f"Error loading or saving the grayscale image: {e}")

# Load DeOldify model
try:
    deoldify_colorizer = get_image_colorizer(artistic=True)  # Load
DeOldify artistic model
    print("DeOldify model loaded successfully.")
except Exception as e:
    raise RuntimeError(f"Error initializing DeOldify: {e}")

# Process image using DeOldify
try:
    colorized_image_path = deoldify_colorizer.plot_transformed_image(
        path=input_path, render_factor=35, watermarked=False
    )
    print(f"Colorized image generated at {colorized_image_path}.")
except Exception as e:
    raise RuntimeError(f"Error during image colorization: {e}")

# Save the colorized image
```

```
try:
    colorized_image = Image.open(colorized_image_path)
    os.makedirs(os.path.dirname(output_path), exist_ok=True)  # Ensure
the directory exists
    colorized_image.save(output_path)
    print(f"Colorized image saved at {output_path}.")
except Exception as e:
    raise IOError(f"Error saving the colorized image: {e}")

print("Colorization with DeOldify completed successfully.")
```

```
Input image saved at /tmp/input_image.jpg.
```

```
/usr/local/lib/python3.10/dist-packages/torch/utils/data/
dataloader.py:617: UserWarning: This DataLoader will create 8 worker
processes in total. Our suggested max number of worker in current
system is 2, which is smaller than what this DataLoader is going to
create. Please be aware that excessive worker creation might get
DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:2
08: UserWarning: The parameter 'pretrained' is deprecated since 0.13
and may be removed in the future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:2
23: UserWarning: Arguments other than a weight enum or `None` for
'weights' are deprecated since 0.13 and may be removed in the future.
The current behavior is equivalent to passing
`weights=ResNet34_Weights.IMAGENET1K_V1`. You can also use
`weights=ResNet34_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
/usr/local/lib/python3.10/dist-packages/torch/nn/utils/weight_norm.py:
143: FutureWarning: `torch.nn.utils.weight_norm` is deprecated in
favor of `torch.nn.utils.parametrizations.weight_norm`.
  WeightNorm.apply(module, name, dim)
/content/DeOldify/fastai/basic_train.py:322: FutureWarning: You are
using `torch.load` with `weights_only=False` (the current default
value), which uses the default pickle module implicitly. It is
possible to construct malicious pickle data which will execute
arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
```

```
related to this experimental feature.
  state = torch.load(tmp_file)
/content/DeOldify/fastai/basic_train.py:271: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
  state = torch.load(source, map_location=device)

DeOldify model loaded successfully.
Colorized image generated at result_images/input_image.jpg.
Colorized image saved at /mnt/data/colorized_output_deoldify.png.
Colorization with DeOldify completed successfully.
```


```
from PIL import Image
import numpy as np
from skimage.color import lab2rgb
from tensorflow.keras.models import load_model

# Load grayscale image
image_path = "/content/dataset/DataSET/Gray/image5001.jpg"  # Replace
with your image path
image = Image.open(image_path).resize((128, 128),
Image.Resampling.LANCZOS)
gray_image = np.array(image.convert("L")) / 255.0
```

```
# Load Autoencoder
autoencoder_model_path =
"/content/colorized_outputs/colorization_autoencoder.h5"
autoencoder = load_model(autoencoder_model_path)

# Preprocess and infer
grayscale_input = gray_image[np.newaxis, :, :, np.newaxis]
predicted_ab = autoencoder.predict(grayscale_input)[0] * 128.0
lab_image = np.concatenate([grayscale_input[0, :, :, 0:1],
predicted_ab], axis=-1)
colorized_image = lab2rgb(lab_image)

# Save result
Image.fromarray((colorized_image *
255).astype(np.uint8)).save("/mnt/data/colorized_output_autoencoder.pn
g")
print("Colorization with Autoencoder completed successfully.")
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have
yet to be built. `model.compile_metrics` will be empty until you train
or evaluate the model.

Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or
evaluate the model.
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Colorization with Autoencoder completed successfully.
```

```
import gradio as gr
import numpy as np
from PIL import Image
import torch
from tensorflow.keras.models import load_model
from skimage.color import lab2rgb
from skimage.metrics import structural_similarity as ssim
from deoldify.visualize import get_image_colorizer
import tempfile


# Load models
class ColorizationModelClass(torch.nn.Module):
    def __init__(self):
        super(ColorizationModelClass, self).__init__()
        self.conv1 = torch.nn.Conv2d(1, 64, kernel_size=3, stride=1,
padding=1)
        self.conv2 = torch.nn.Conv2d(64, 128, kernel_size=3, stride=1,
padding=1)
        self.conv3 = torch.nn.Conv2d(128, 3, kernel_size=3, stride=1,
padding=1)
```

```
    def forward(self, x):
        x = torch.relu(self.conv1(x))
        x = torch.relu(self.conv2(x))
        x = torch.sigmoid(self.conv3(x))
        return x


# Model paths
model1_path = "/content/dataset/colorization_model.pth"
deoldify_model_path =
"/content/dataset/ColorizeArtistic_gen_saved.pth"
autoencoder_model_path =
"/content/colorized_outputs/colorization_autoencoder.h5"

# Load Model 1 (PyTorch)
try:
    model1 = ColorizationModelClass()
    model1.load_state_dict(torch.load(model1_path,
map_location=torch.device("cpu")), strict=False)  # Use strict=False
to ignore missing keys
    model1.eval()
except Exception as e:
    raise RuntimeError(f"Error loading PyTorch model: {e}")

# Load Model 2 (DeOldify)
try:
    deoldify_colorizer = get_image_colorizer(artistic=True)
except Exception as e:
    raise RuntimeError(f"Error initializing DeOldify: {e}")

# Load Model 3 (Autoencoder)
try:
    autoencoder = load_model(autoencoder_model_path)
except Exception as e:
    raise RuntimeError(f"Error loading Autoencoder model: {e}")


# Preprocessing
def preprocess_image(image: Image.Image, size: int = 128) ->
np.ndarray:
    """Resize and normalize grayscale image."""
    gray_image = image.convert("L").resize((size, size),
Image.Resampling.LANCZOS)
    return np.array(gray_image) / 255.0


# Colorization Functions
def colorize_model1(image: Image.Image) -> Image.Image:
    """Colorize using PyTorch model."""
    try:
```

```
        gray_image = preprocess_image(image)
        input_tensor = torch.tensor(gray_image[np.newaxis, np.newaxis,
:, :], dtype=torch.float32)
        with torch.no_grad():
            output_tensor = model1(input_tensor).numpy()
        # Adjust scaling
        colorized_image = (output_tensor[0].transpose(1, 2, 0) *
255.0).clip(0, 255).astype(np.uint8)
        return Image.fromarray(colorized_image)
    except Exception as e:
        raise RuntimeError(f"Error in PyTorch model colorization:
{e}")


def colorize_deoldify(image: Image.Image) -> Image.Image:
    """Colorize using DeOldify model."""
    try:
        with tempfile.NamedTemporaryFile(suffix=".jpg", delete=False)
as input_file:
            image.save(input_file.name)
            colorized_path =
deoldify_colorizer.plot_transformed_image(
                path=input_file.name, render_factor=35,
watermarked=False
            )
        return Image.open(colorized_path)
    except Exception as e:
        raise RuntimeError(f"Error in DeOldify model colorization:
{e}")


def colorize_autoencoder(image: Image.Image) -> Image.Image:
    """Colorize using Autoencoder model."""
    try:
        gray_image = preprocess_image(image)
        grayscale_input = gray_image[np.newaxis, :, :, np.newaxis]
        # Ensure proper scaling and denormalization
        predicted_ab = autoencoder.predict(grayscale_input)[0] * 128.0
        lab_image = np.concatenate([grayscale_input[0, :, :, :1] *
100.0, predicted_ab], axis=-1)  # L channel rescaled
        colorized_image = lab2rgb(lab_image)
        return Image.fromarray((colorized_image * 255).clip(0,
255).astype(np.uint8))
    except Exception as e:
        raise RuntimeError(f"Error in Autoencoder model colorization:
{e}")


# Metrics Calculation
def calculate_metrics(original: Image.Image, colorized: Image.Image) -
```

```
> str:
    """Calculate SSIM, MSE, and PSNR for grayscale versions of the
images."""
    # Ensure both images are the same size
    original_resized = original.resize(colorized.size,
Image.Resampling.LANCZOS)

    # Convert to grayscale
    original_gray = np.array(original_resized.convert("L"))
    colorized_gray = np.array(colorized.convert("L"))

    # SSIM
    ssim_value = ssim(original_gray, colorized_gray, data_range=255)

    # MSE
    mse_value = np.mean((original_gray - colorized_gray) ** 2)

    # PSNR
    psnr_value = 100 if mse_value == 0 else 20 * np.log10(255.0 /
np.sqrt(mse_value))

    return f"SSIM: {ssim_value:.4f}\nMSE: {mse_value:.4f}\nPSNR:
{psnr_value:.4f}"


# Gradio Interface
def colorize_image(model_name: str, grayscale_image: Image.Image):
    if not grayscale_image:
        return None, "Error: Please upload a valid grayscale image."

    try:
        if model_name == "PyTorch Model":
            colorized_image = colorize_model1(grayscale_image)
        elif model_name == "DeOldify Model":
            colorized_image = colorize_deoldify(grayscale_image)
        elif model_name == "Autoencoder Model":
            colorized_image = colorize_autoencoder(grayscale_image)
        else:
            raise ValueError("Invalid model name selected.")

        metrics = calculate_metrics(grayscale_image, colorized_image)
        return colorized_image, metrics
    except Exception as e:
        return None, f"Error: {e}"


# Gradio App
model_names = ["PyTorch Model", "DeOldify Model", "Autoencoder Model"]
interface = gr.Interface(
    fn=colorize_image,
```

```
    inputs=[
        gr.Radio(choices=model_names, label="Choose Colorization
Model"),
        gr.Image(label="Upload Grayscale Image", type="pil"),
    ],
    outputs=[
        gr.Image(label="Colorized Image"),
        gr.Textbox(label="Metrics", interactive=False),
    ],
    title="Enhanced Image Colorization Dashboard",
    description=(
        "Choose a colorization model, upload a grayscale image, "
        "and view the colorized result with SSIM, MSE, and PSNR
metrics."
    ),
    theme="default",
)

# Launch
interface.launch()
```

```
<ipython-input-24-5b2be00c8cb3>:35: FutureWarning: You are using
`torch.load` with `weights_only=False` (the current default value),
which uses the default pickle module implicitly. It is possible to
construct malicious pickle data which will execute arbitrary code
during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  model1.load_state_dict(torch.load(model1_path,
map_location=torch.device("cpu")), strict=False)  # Use strict=False
to ignore missing keys
WARNING:absl:Compiled the loaded model, but the compiled metrics have
yet to be built. `model.compile_metrics` will be empty until you train
or evaluate the model.

Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or
evaluate the model.

INFO:httpx:HTTP Request: GET https://api.gradio.app/pkg-version
"HTTP/1.1 200 OK"
```

```
HTTP Request: GET https://api.gradio.app/pkg-version "HTTP/1.1 200 OK"

INFO:httpx:HTTP Request: GET http://127.0.0.1:7860/gradio_api/startup-events "HTTP/1.1 200 OK"

HTTP Request: GET http://127.0.0.1:7860/gradio_api/startup-events "HTTP/1.1 200 OK"

INFO:httpx:HTTP Request: HEAD http://127.0.0.1:7860/ "HTTP/1.1 200 OK"

Running Gradio in a Colab notebook requires sharing enabled. Automatically setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

HTTP Request: HEAD http://127.0.0.1:7860/ "HTTP/1.1 200 OK"

INFO:httpx:HTTP Request: GET https://api.gradio.app/v3/tunnel-request "HTTP/1.1 200 OK"

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
HTTP Request: GET https://api.gradio.app/v3/tunnel-request "HTTP/1.1 200 OK"

INFO:httpx:HTTP Request: GET https://cdn-media.huggingface.co/frpc-gradio-0.3/frpc_linux_amd64 "HTTP/1.1 200 OK"

HTTP Request: GET https://cdn-media.huggingface.co/frpc-gradio-0.3/frpc_linux_amd64 "HTTP/1.1 200 OK"

INFO:httpx:HTTP Request: HEAD https://36c74d1cb8b84984d3.gradio.live "HTTP/1.1 200 OK"

* Running on public URL: https://36c74d1cb8b84984d3.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)
HTTP Request: HEAD https://36c74d1cb8b84984d3.gradio.live "HTTP/1.1 200 OK"

<IPython.core.display.HTML object>
```

```
from pyngrok import ngrok, conf
import os

# Set the ngrok auth token
conf.get_default().auth_token = "2phP00Avhicmtseo8jiNS52EH7C_7zSVMRrbX8hTNNWcjpjet"
```

```
# Run the Streamlit app
os.system(r"streamlit run C:\Users\altaf\Desktop\Colorisation\
colorization_env\dashboard.py &")

# Connect ngrok to the correct port
public_url = ngrok.connect(8501)  # 8501 is the default Streamlit port
print("Public URL:", public_url)
```

```
INFO:pyngrok.ngrok:Opening tunnel named: http-8501-4560f1c6-3b00-451a-
a507-31ec7b8e910a

Opening tunnel named: http-8501-4560f1c6-3b00-451a-a507-31ec7b8e910a

INFO:pyngrok.process.ngrok:t=2024-12-03T20:31:07+0000 lvl=info
msg=start pg=/api/tunnels id=e64487fd613ddcd2

t=2024-12-03T20:31:07+0000 lvl=info msg=start pg=/api/tunnels
id=e64487fd613ddcd2

INFO:pyngrok.process.ngrok:t=2024-12-03T20:31:07+0000 lvl=info
msg="started tunnel" obj=tunnels name=http-8501-4560f1c6-3b00-451a-
a507-31ec7b8e910a addr=http://localhost:8501 url=https://a2e8-34-83-
23-10.ngrok-free.app

Public URL: NgrokTunnel: "https://a2e8-34-83-23-10.ngrok-free.app" ->
"http://localhost:8501"
t=2024-12-03T20:31:07+0000 lvl=info msg="started tunnel" obj=tunnels
name=http-8501-4560f1c6-3b00-451a-a507-31ec7b8e910a
addr=http://localhost:8501 url=https://a2e8-34-83-23-10.ngrok-free.app

INFO:pyngrok.process.ngrok:t=2024-12-03T20:31:07+0000 lvl=info msg=end
pg=/api/tunnels id=e64487fd613ddcd2 status=201 dur=69.369332ms

t=2024-12-03T20:31:07+0000 lvl=info msg=end pg=/api/tunnels
id=e64487fd613ddcd2 status=201 dur=69.369332ms
```