

DANISH RYWEN T

25MML0048

MACSE604 – EDGE INTELLIGENCE

Error: unzip is not recognized as an internal or external command

While extracting the dataset, the Linux-based unzip command was used in a Windows Jupyter Notebook environment.

```
[1]: !unzip Animals.zip

'unzip' is not recognized as an internal or external command,
operable program or batch file.

[ ]:
```

The unzip command works only in Linux/Google Colab environments. Windows Command Prompt and Jupyter Notebook do not support this command.

```
[1]: import zipfile
from pathlib import Path

zip_path = Path(r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals.zip")
extract_to = Path(r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_extracted")

extract_to.mkdir(parents=True, exist_ok=True)

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

print("Extracted to:", extract_to)
```

Extracted to: C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_extracted

Error : Image Distortion Due to Direct Resizing

What was the Error?

Initially, images were resized directly to 224×224 pixels using the `resize()` function without preserving the original aspect ratio.

What was Actually Happening?

Images with different height-width ratios were **stretched or squashed**, causing distortion in visual content.

This distortion negatively affects feature extraction and reduces the performance of machine learning models.

How I Identified the Error

After visualizing the resized images, it was observed that objects appeared elongated or compressed compared to the original images.

```
import os
from PIL import Image, ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True

DATA_DIR = r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_extracted"
RESIZED_DIR = r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_resized_wrong"
TARGET_SIZE = (224, 224)

os.makedirs(RESIZED_DIR, exist_ok=True)

for root, dirs, files in os.walk(DATA_DIR):
    for f in files:
        if f.lower().endswith(("jpg", ".jpeg", ".png")):
            src = os.path.join(root, f)

            rel = os.path.relpath(root, DATA_DIR)
            out_folder = os.path.join(RESIZED_DIR, rel)
            os.makedirs(out_folder, exist_ok=True)

            dst = os.path.join(out_folder, f)

            try:
                img = Image.open(src).convert("RGB")

                img = img.resize(TARGET_SIZE)

                img.save(dst)
            except:
                pass

print("Resizing completed!")
```

How I Rectified the Error

A **pad-to-square** technique was introduced before resizing.

This method adds padding to the shorter side of the image, making it square while preserving aspect ratio.

```
[3]: import os
from PIL import Image, ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True

DATA_DIR = r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_extracted"
RESIZED_DIR = r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_resized"
TARGET_SIZE = (224, 224)

os.makedirs(RESIZED_DIR, exist_ok=True)

def pad_to_square(img):
    w, h = img.size
    if w == h:
        return img
    s = max(w, h)
    new = Image.new("RGB", (s, s), (0,0,0))
    new.paste(img, ((s-w)//2, (s-h)//2))
    return new

for root, dirs, files in os.walk(DATA_DIR):
    for f in files:
        if f.lower().endswith(("jpg", "jpeg", ".png")):
            src = os.path.join(root, f)

            rel = os.path.relpath(root, DATA_DIR)
            out_folder = os.path.join(RESIZED_DIR, rel)
            os.makedirs(out_folder, exist_ok=True)

            dst = os.path.join(out_folder, f)

            try:
                img = Image.open(src).convert("RGB")
                img = pad_to_square(img)
                . . .
```

Excessive Console Output During Feature Extraction

The feature extraction loop initially processed and printed results for all images in the dataset.

What was Actually Happening?

Printing features for thousands of images overwhelmed the console and made debugging difficult.

How I Identified the Error

The notebook became slow and cluttered with excessive output during execution.

How I Overcame the Error

I limited the number of images processed using a counter:

MAX_IMAGES_TO_PROCESS = 10

This allowed controlled output and easier verification of extracted features.

```
import os
import numpy as np
from PIL import Image, ImageFile
import cv2

ImageFile.LOAD_TRUNCATED_IMAGES = True

RESIZED_DIR = r"C:\Users\danis\EDGE INTELLIGENCE LAB\Animals_resized"

image_count = 0
MAX_IMAGES_TO_PROCESS = 10 # limit output

for root, dirs, files in os.walk(RESIZED_DIR):
    for f in files:
        if f.lower().endswith('.jpg', '.jpeg', '.png'):
            path = os.path.join(root, f)

            try:
                img = Image.open(path).convert("RGB")
                arr = np.array(img)
            except Exception as e:
                print(f"Error opening image {path}: {e}")
                continue
```

VISUALIZATION:

