THE DOT PY HELPER FUNCTIONS CODE BELOW

from pathlib import Path

import cv2, numpy as np, pandas as pd

ROOT = Path.cwd()

DATA\_DIR = ROOT / "BRIDGE CORROSION VDOT"

OUTPUT\_DIR = Path("output"); OUTPUT\_DIR.mkdir(parents=True, exist\_ok=True)

DATASET\_CSV = OUTPUT\_DIR / "capstone\_dataset.csv"

DATASET\_XLSX = OUTPUT\_DIR / "capstone\_dataset.xlsx"

TARGET\_SIZE = (1024, 768)

EXTS = {".jpg", ".jpeg", ".png", ".tif", ".tiff"}

def imread\_any(path: str):

img = cv2.imread(str(path), cv2.IMREAD\_COLOR)

if img is None:

raise FileNotFoundError(f"Could not load image: {path}")

return cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

def ensure\_size(img, wh):

w, h = wh

return cv2.resize(img, (w, h), interpolation=cv2.INTER\_AREA)

def to\_gray(img):

return cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)

def quick\_edge\_length(gray):

edges = cv2.Canny(gray, 80, 160)

kernel = np.ones((3,3), np.uint8)

thin = cv2.morphologyEx(edges, cv2.MORPH\_OPEN, kernel, iterations=1)

return float(np.count\_nonzero(thin))

def quick\_rust\_area(rgb):

hsv = cv2.cvtColor(rgb, cv2.COLOR\_RGB2HSV)

lower = np.array([5, 50, 20], dtype=np.uint8)

upper = np.array([25, 255, 255], dtype=np.uint8)

mask = cv2.inRange(hsv, lower, upper)

kernel = np.ones((5,5), np.uint8)

mask = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel, iterations=1)

return float(np.count\_nonzero(mask))

def px\_to\_inches(px, dpi=96.0): return px / dpi

LOW\_TAGS = ("LOW",)

MED\_TAGS = ("MED", "MEDIUM")

SEV\_TAGS = ("SEV", "SEVERE")

def find\_one(folder: Path, tags):

for p in sorted(folder.iterdir()):

if p.is\_file() and p.suffix.lower() in EXTS and any(t in p.stem.upper() for t in tags):

return p

raise FileNotFoundError(f"No file with tags {tags} found in {folder}")

def main():

folders = sorted([p for p in DATA\_DIR.glob("IMG \*") if p.is\_dir()], key=lambda x: x.name)

rows = []

for fld in folders:

p\_low = find\_one(fld, LOW\_TAGS); p\_med = find\_one(fld, MED\_TAGS); p\_sev = find\_one(fld, SEV\_TAGS)

for label, p in [("Low", p\_low), ("Medium", p\_med), ("Severe", p\_sev)]:

img = ensure\_size(imread\_any(p), TARGET\_SIZE)

gray = to\_gray(img)

L = quick\_edge\_length(gray)

A = quick\_rust\_area(img)

rows.append({

"Bridge\_ID": fld.name,

"Image\_Path": str(p),

"Risk\_Label": label,

"Crack\_Length\_in": px\_to\_inches(L),

"Rust\_Area\_in2": A/(96.0\*96.0)

})

df = pd.DataFrame(rows)

df.to\_csv(DATASET\_CSV, index=False)

df.to\_excel(DATASET\_XLSX, index=False)

print("Wrote:", DATASET\_CSV.resolve())

print("Wrote:", DATASET\_XLSX.resolve())

if \_\_name\_\_ == "\_\_main\_\_":

main()