[1] Notebook — Cell #1

Tool: ChatGPT
Prompt: “Write code to load JPEG images into TensorFlow, convert to RGB, normalize, and prepare them for a train/val/test split.”
Output Used: Dataset loading pipeline, image decoding, normalization code.

---

[2] Notebook — Cell #2

Tool: ChatGPT
Prompt: “Create a stratified 70/15/15 split function for my damage/no_damage folders using TensorFlow or Python.”
Output Used: `split_per_class()` function with stratification logic.

---

[3] Notebook — Cell #3

Tool: ChatGPT
Prompt: “Help me create a tf.data pipeline with batching, shuffling, and prefetch.”
Output Used: Code for dataset batching, prefetching, and shuffling.

---

[4] Notebook — Cell #4

Tool: ChatGPT
Prompt: “Implement a dense baseline neural network for 128×128×3 images.”
Output Used: Dense model architecture.

---

[5] Notebook — Cell #5

Tool: ChatGPT
Prompt: “Implement LeNet-5 in TensorFlow for my resolution.”
Output Used: Modified LeNet-5 architecture code.

---

[6] Notebook — Cell #6

Tool: ChatGPT
Prompt: “Implement the Alternate-LeNet architecture from the 2018 paper (Table 1).”
Output Used: Alternate-LeNet model architecture.

---

[7] Notebook — Cell #7

Tool: ChatGPT
Prompt: “Write the training code for all three models and return history curves.”
Output Used: `.fit()` training loops with callbacks and class weights.

---

[8] Notebook — Cell #8

Tool: ChatGPT
Prompt: “Write evaluation code to compute accuracy, confusion matrix, classification report, ROC-AUC, and per-model scores.”
Output Used: Evaluation script and metrics.

---

[9] Notebook — Cell #9

Tool: ChatGPT
Prompt: “Write code to save the model with model.save('saved_models/best_model.keras').”
Output Used: Save/export code.

---

[10] Notebook — Cell #10

Tool: ChatGPT
Prompt: “Plot confusion matrix and ROC.”
Output Used: Matplotlib/Sklearn plotting code.

---

[11] Notebook — Cell #11

Tool: ChatGPT
Prompt: “Compare validation accuracy of dense, lenet5, alt_lenet and return dictionary of results.”
Output Used: Model comparison snippet.

---

[12] Notebook — Cell #12

Tool: ChatGPT
Prompt: “Automatically print model summaries and parameter counts for all three models.”
Output Used: Model summary loop.

Inference Server (Part 3)

[13] server/app.py

Tool: ChatGPT
Prompt: “Write a Flask server with GET /summary and POST /inference endpoints that accepts raw binary JPEG and returns {prediction: damage/no_damage}.”
Output Used: Entire Flask app including preprocessing and model loading.

---

[14] server/requirements.txt

Tool: ChatGPT
Prompt: “Generate minimal requirements.txt for TensorFlow + Flask inference server.”
Output Used: Requirements list.

---

[15] Dockerfile

Tool: ChatGPT
Prompt: “Write a Dockerfile that runs my Flask inference server and loads my Keras model.”
Output Used: Dockerfile contents.

---

[16] docker-compose.yml

Tool: ChatGPT
Prompt: “Create docker-compose.yml that maps port 5000 and mounts saved_models.”
Output Used: docker-compose spec.