INITIAL PROMPT:

ok so like previously i was making the organic/inorganic model with sepreate head for classification and quality regression and then added a yolo pretrained yolo model to help identify the objects, however i am not satisfied in using supervised and unsupervised as the unsupervised is taking as lot of wrong data and yolo isnt integrating within properly. Instead I will recreate the entire thing however we will only use supervised learning i.e the model takes and learns from the given database instead. The model would give an output as such:

┌────────────────────────────┬────────────────────────────┬────────────────────────────┐

│ Original Image (with │ Classification Grad-CAM │ Quality Grad-CAM │

│ YOLO Boxes drawn) │ │ │

├────────────────────────────┼────────────────────────────┼────────────────────────────┤

│ 🔍 PREDICTION │ 🧠 CLASSIFICATION │ 🎯 QUALITY │

│ - Label, Confidence │ EXPLANATION │ EXPLANATION │

│ - Quality Score, Grade │ - Color Var, Shape, etc. │ - Color Var, Shape, etc. │

│ - YOLO objects: count │ - Dark Spots │ - Dark Spots │

│ │ │ │

└────────────────────────────┴────────────────────────────┴────────────────────────────┘

Also the database would be on a limited dataset of a certain number of fruits but will include all details of available images possible such as organic inorganic good quality medium quality bad quality or as good surface ebad surface idk yet. Basically since we are starting from scratch guide on how the dataset should be made and what details it may require to visually differentiate between an organic and an inorganic fruit and likewise give an idea of the quality of the frit based on size irregularity colour shininess surface texture dark spots or whatever other requirements that may be needed and help visually differentiate and then give an output. Also since the output must also include along with the number of objects detected which needs to be accurate also be able to tell the number of shiny fruits the number of big medium small sized fruits based on the average size of the fruit shown in the photo number of shiny fruits as well etc

So guide how the dataset should be structured and what type of photos I may require to build an efficient model with high accuracy and give the desired output. But first also see what visual differences can be used to give the complete correct output j using visual pictures and then give the outputsss

**A. Organic vs Inorganic – Key Visual Cues**

| **Feature** | **Organic Fruit** | **Inorganic Fruit** |
| --- | --- | --- |
| **Surface Texture** | More irregular, natural blemishes | Very smooth, polished |
| **Color Consistency** | Varied shades, occasional uneven ripening | Uniform color, waxed appearance |
| **Shininess** | Low to moderate, natural dullness | Highly shiny (wax or preservatives) |
| **Shape** | Slight asymmetries, natural imperfections | Near-perfect symmetry |
| **Dark Spots** | More likely due to natural growth | Less likely; usually cleaned or filtered out |

**B. Quality – Key Visual Cues**

| **Feature** | **High Quality** | **Medium Quality** | **Low Quality** |
| --- | --- | --- | --- |
| **Size** | Large, consistent | Medium, inconsistent | Small or overly large |
| **Shininess** | Slight shine (natural moisture) | Dull or excessive shine | Dull, dry, or dirty |
| **Color** | Vibrant and fresh | Slight discoloration | Dark patches or fading |
| **Dark Spots** | Minimal or none | Some spotting | Excessive or mold-like |
| **Shape** | Symmetrical or near-ideal | Minor irregularities | Misshapen or dented |

EXAMPLE OUTPUT:

┌────────────────────────────┬────────────────────────────┬────────────────────────────┐

│ Original Image (w/ boxes) │ Classification Grad-CAM │ Quality Grad-CAM │

├────────────────────────────┼────────────────────────────┼────────────────────────────┤

│ 🔍 PREDICTION │ 🧠 CLASSIFICATION │ 🎯 QUALITY │

│ - Label: Organic │ - Shape asymmetry detected │ - Smooth, glossy surface │

│ - Confidence: 94.3% │ - Uneven color tones │ - No dark spots detected │

│ - Quality: Medium (0.68) │ - Dull shine │ - Some wrinkling visible │

│ - Objects: 3 │ │ │

│ - Big: 1, Med: 2, Small: 0 │ │ │

│ - Shiny: 2 │ │ │

└────────────────────────────┴────────────────────────────┴────────────────────────────┘

DATASET:

your\_dataset/

├── images/

│ ├── apple\_001.jpg

│ ├── banana\_002.jpg

│ └── ...

├── labels.csv ✅ Master label file

└── metadata/

└── gradcam\_masks/ (optional, for debugging or later overlays)

Labels.csv

| **image\_name** | **fruit\_type** | **is\_organic** | **quality\_grade** | **size** | **shininess** | **dark\_spots** | **shape\_irregularity** | **notes** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| apple\_001.jpg | apple | 1 | good (2) | large | shiny | 0 | low | - |
| mango\_002.jpg | mango | 0 | bad (0) | small | dull | 1 | high | mold patches |
| banana\_003.jpg | banana | 1 | medium (1) | med | shiny | 1 | med | uneven peel |

| **Output Head** | **Task Type** | **Label(s) Used From CSV** |
| --- | --- | --- |
| organic\_classifier | Binary Class | is\_organic |
| quality\_regressor | Regression | quality\_grade (0–2) |
| size\_classifier | Categorical | size (small/medium/large) |
| shininess\_classifier | Binary Class | shininess (shiny/dull) |
| darkspot\_detector | Binary Class | dark\_spots (0/1) |
| shape\_irregularity | Regression or Class | shape\_irregularity (0–1 or class) |

| **Feature** | **Format for Model** | **Encoder Example** |
| --- | --- | --- |
| is\_organic | 0 = inorganic, 1 = organic | BinaryEncoder |
| quality\_grade | 0 = bad, 1 = medium, 2 = good | OrdinalEncoder or Integer |
| size | one-hot or label encoded | small=0, med=1, large=2 |
| shininess | 0 = dull, 1 = shiny | BinaryEncoder |
| dark\_spots | 0 = no, 1 = yes | BinaryEncoder |
| shape\_irregularity | Regression (0–1) or Class | float from 0 to 1 |

ROUGH EXAMPLE:

FRUIT-CLASSIFY-QUALITY-DETECTOR/

├── scripts/

│ ├── train.py ✅ Main training script

│ ├── dataset\_loader.py ✅ tf.data loader, label parsing

│ ├── model\_builder.py ✅ builds the multi-head CNN model

│ ├── loss\_metrics.py ✅ defines custom losses/metrics (if any)

│ ├── gradcam\_utils.py ✅ Grad-CAM logic per output head

│ ├── predict\_and\_explain.py ✅ takes image → formatted Grad-CAM output

│ └── config.py ✅ constants (paths, input size, class labels)

│

├── data/

│ ├── images/

│ ├── labels.csv

│ └── test\_images/

├── outputs/

│ ├── model\_weights/

│ ├── logs/

│ └── predictions/

TEN COMMON FRUITS USED:

 **Mango (Aam)**

 **Banana (Kela)**

 **Apple (Saib)**

 **Orange (Malta / Kinnow)**

 **Guava (Amrood)**

 **Pomegranate (Anar)**

 **Grapes (Angoor)**

 **Watermelon (Tarbooz)**

 **Papaya (Papita)**

 **Strawberries**

(WILL ALSO INCLUE A NOTES WHICH HAS BATCH OR SINGLE WRITTEN BASED ON PHOTO)