# chatsonic

Sure, here is an example code for apple maturity detection using CNN and YOLOv8 in Python programming language:
```python
# Import necessary libraries
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import load\_model
# Load YOLOv8 model
net = cv2.dnn.readNet("yolov8.weights", "yolov8.cfg")
# Define classes for apple and background
classes = ["apple", "background"]
# Load CNN model for maturity detection
model = load\_model("apple\_maturity\_detection.h5")
# Define function for detecting apple maturity
def detect\_maturity(image):
# Convert image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)
# Detect apple using YOLOv8
blob = cv2.dnn.blobFromImage(image, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
outs = net.forward(net.getUnconnectedOutLayersNames())
class\_ids = []
confidences = []
boxes = []
for out in outs:
for detection in out:
scores = detection[5:]
class\_id = np.argmax(scores)
confidence = scores[class\_id]
if confidence > 0.5 and class\_id == 0:
center\_x = int(detection[0] \* image.shape[1])
center\_y = int(detection[1] \* image.shape[0])
w = int(detection[2] \* image.shape[1])
h = int(detection[3] \* image.shape[0])
x = center\_x - w / 2
y = center\_y - h / 2
boxes.append([x, y, w, h])
confidences.append(float(confidence))
class\_ids.append(class\_id)
# Apply non-max suppression to remove overlapping boxes
indices = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
# Loop over detected apple boxes
for i in indices:
i = i[0]
box = boxes[i]
x, y, w, h = box
# Extract apple region from image
apple\_img = gray[int(y):int(y+h), int(x):int(x+w)]
# Resize apple region to match CNN input shape
apple\_img\_resized = cv2.resize(apple\_img, (100, 100))
# Normalize apple region image
apple\_img\_norm = apple\_img\_resized / 255.0
# Reshape apple region image for CNN input
apple\_img\_reshaped = apple\_img\_norm.reshape((1, 100, 100, 1))
# Predict maturity using CNN model
maturity = model.predict(apple\_img\_reshaped)
# Display prediction on image
label = "Mature" if maturity > 0.5 else "Immature"
cv2.putText(image, label, (int(x), int(y) - 5), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)
# Return image with maturity predictions
return image
# Test the function on an example image
image = cv2.imread("apple.jpg")
output = detect\_maturity(image)
cv2.imshow("Output", output)
cv2.waitKey(0)
cv2.destroyAllWindows()
```
This code uses YOLOv8 to detect the apple in an input image and then uses a CNN model to predict the maturity of the detected apple. The output image displays the predicted maturity label on the detected apple. Note that you will need to train your own CNN model for apple maturity detection using a dataset of mature and immature apple images.