Stap 1: Wat is het probleem?
Mijn machine learning model moet via een raspberry pi en een camera module m&ms hun kleur identificeren dit kan met opencv maar ik krijg extra punten als ik het met een machine learning model doe. Ook maakt opencv verkeerde waarnemingen door lichtomstandigheden en is het fouten marge te groot. Ik kan de m&m ook niet volledig isoleren dus ik ben bang dat opencv de achtergrond ook als m&m kleur zou kunnen omschrijven.

Stap 2: welke data heb ik nodig?
Mijn plan is om een model te maken dat de kleur van verschillende objecten kan detecteren maar hiervoor heb ik een heel uitgebreide database nodig. Ik zou kunnen kijken naar datasets met objecten die van kleur vergelijkbaar zijn met die van een m&m maar ik weet niet goed of mijn model de m&m dan als een object zal herkennen. Ook kan ik een bestaande dataset labelen volgens de m&m kleuren. Mijn andere optie is zelf een databank maken van alle m&m's en hun kleuren in verschillende omstandigheden. Voor de dataset heb ik een dataset nodig die verschillende lichtomstandigheden en hoeken van het object of de m&m omvat. Dit maakt mijn model robuuster verkleint de foutenmarge

Stap 3: data verkrijgen.
Ik heb drie opties om de data te verkrijgen. Op het internet zoeken naar een dataset of zelf een dataset maken. Op het internet zal ik waarschijnlijk geen dataset vinden specifiek rondt de kleuren van m&m snoepjes daarom dacht ik dat het ook mogelijk zou zijn om een machine learning model te maken die van objecten de kleur kan detecteren. Mijn andere optie is om zelf een dataset te maken door zelf foto's te nemen of foto's te zoeken op het internet. Op google vindt ik echter niet genoeg geschikte foto's om mijn model te trainen. Dat probleem kan ik oplossen door mijn foto's op de computer te bewerken (rotatie, helderheid, contrast en zoom aanpassen) Zelf een dataset maken is een mogelijkheid maar veel werk. Als ik zelf een dataset maak moet ik nog al mijn data labelen met bv labellmg of automatisch met opencv.

Stap 4: het model zijn mogelijkheden bepalen en het model trainen.
Het model moet instaat zijn om foutloos alle kleuren van m&m's te herkennen en deze vervolgens sorteren. Om dit goed te doen moet ik kiezen hoe het model zichzelf traint, een algoritme kiezen.(uit mijn research lijkt een convolutional neuraal netwerk de best optie en op de raspberry pi's werken Resnet en Mobilenet goed), het model trainen om dit te doen moet ik eerst de grootte en variatie van mijn dataset weten zo kan ik bepalen hoeveel epochs(keren dat het model door de trainingsdata gaat)het model moet doen dit mag echter niet teveel zijn anders herkent het model patronen in de trainingsdata maar niet in nieuwe afbeeldingen dit noemt overfitting. Gezien de complexiteit van het project zou mijn model afhankelijk van de grote van mijn dataset 10-50 epochs moeten doorlopen. Om niet handmatig te moeten "gokken" hoeveel epochs ik nodig heb kan ik een code maken die het model vanzelf stopt met de dataset te doornemen dit doe het door middel van een validatieset na elke epoch. Het model berekent dan de nauwkeurigheid en zolang deze stijgt gaat het model door met trainen als de nauwkeurigheid zoveel keer na elkaar niet stijgt (deze waarde stel je vooraf in) dan reset het model zichzelf naar de beste versie(de nauwkeurigste). Je stelt dan op voorhand gewoon een max aantal epochs in. Na die training moet het model zijn parameters ingesteld hebben en zichzelf valideren.

Stap 5: model evalueren
deze stap bevat:het model evalueren, confusion matrix berekenen in mijn geval is dit een multiclass confusion matrix aangezien er meerder kleuren m&m's zijn(deze berekeningen haal je uit een tabel die de prestaties van het model weergeeft in TP-True Positive het model voorspelde positief en de werkelijke waarde was positief, TN-True Negative het model voorspelde negatief en het was negatief, FP-False Positive het model voorspelde positief maar het was negatief, FN-False Negative het model voorspelde negatief maar het was positief). Uit deze berekeningen kun je volgende waardes halen nauwkeurigheid, precisie,gevoeligheid en F1-score(formules hiervoor vindt je op het eind van het document) Kpi's,
model performance metric, kwaliteit van het model evalueren, eindtest(voldoet het model aan mijn eisen)

Stap 6: experimenteren en aanpassen als het model in werking is.
In deze stap moet ik het model toepassen op mijn project (de m&m sorteerder) en het model nog proberen verbeteren. Verbeteren kun je doen door de data waar het model nog over struikelt toe te voegen in de trainingsdata.

Stap 7: Het proces herhalen en verbeteren
In deze stap is het de bedoeling dat ik het volledige proces nog eens doorloop en mijn model verbeter.Dit betekent dus de trainingsdata etc aanpassen indien nodig. Om dit goed te bij te houden kan ik tools zoals tensorflow gebruiken om de verbetering van mijn model te documenteren.

Formules:

$$nauwkeurigheid = \frac{TP+TN}{TP+TN+FP+FN}$$
Het percentage correcte voorspellingen (zowel TP als TN) over het totaal aantal voorspellingen.

$$precisie = \frac{TP}{TP+FP}$$
Hoe vaak de positieve voorspelling correct is (van alle keren dat het model "positief" voorspelde, hoe vaak was dat juist).

$$gevoeligheid = \frac{TP}{TP+FN}$$
Hoe goed het model in staat is om de echte positieve gevallen te herkennen (van alle echte positieve gevallen, hoe vaak voorspelde het model correct).

$$f1\ score = 2 \times \frac{precisie \times gevoeligheid}{precisie + gevoeligheid}$$
Een samengestelde score die de balans weergeeft tussen precisie en gevoeligheid. Het is handig wanneer er een onbalans is tussen TP, FP, en FN.