Edge Impulse op een ESP32-CAM - Image Classification

Inhoudsopgave

1. Inleiding

Dit document beschrijft hoe je Edge Impulse kunt gebruiken met een ESP32-CAM voor posture detection of een ander classificatie model. We doorlopen stap voor stap het proces van setup tot deployment van het model.

2. Benodigde Software

- Arduino IDE: https://www.arduino.cc/en/software/
- Edge Impulse (Online Omgeving): https://edgeimpulse.com

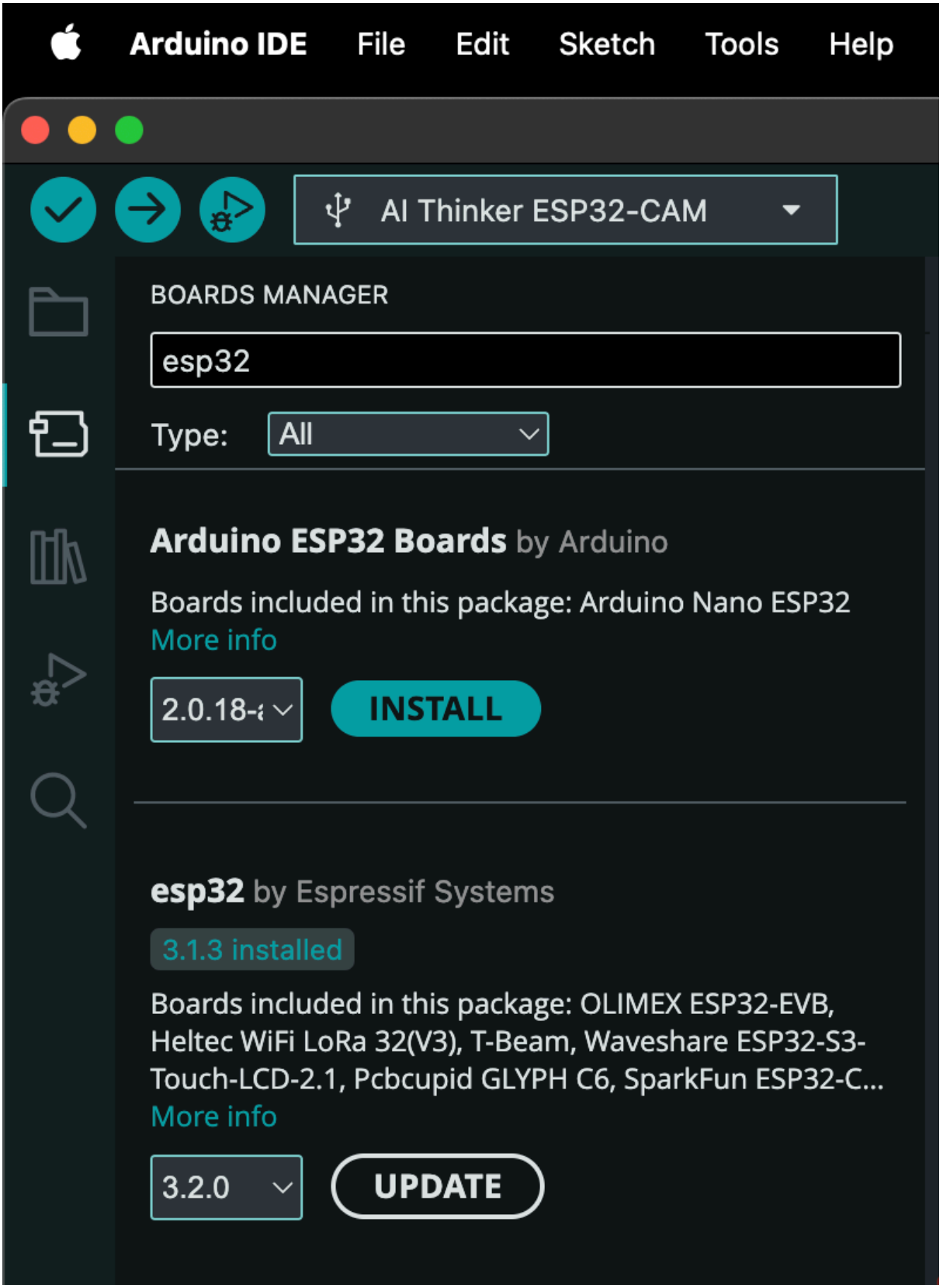
3. Benodigde Hardware

- ESP32-CAM met een OV2640 Camera module (BELANGRIJK!)

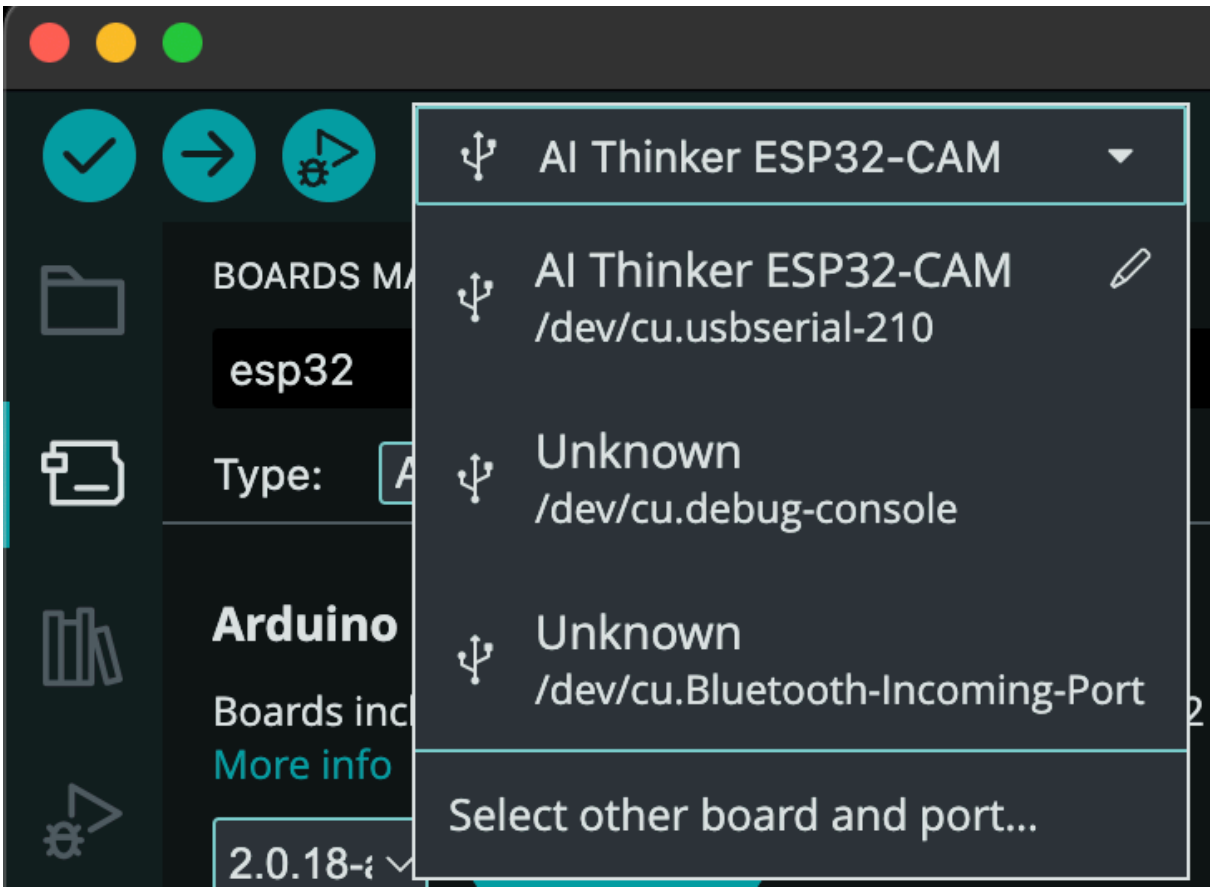
- ESP32-CAM-MB Programmer Shield
- Micro-USB Kabel

4. IDE Setup

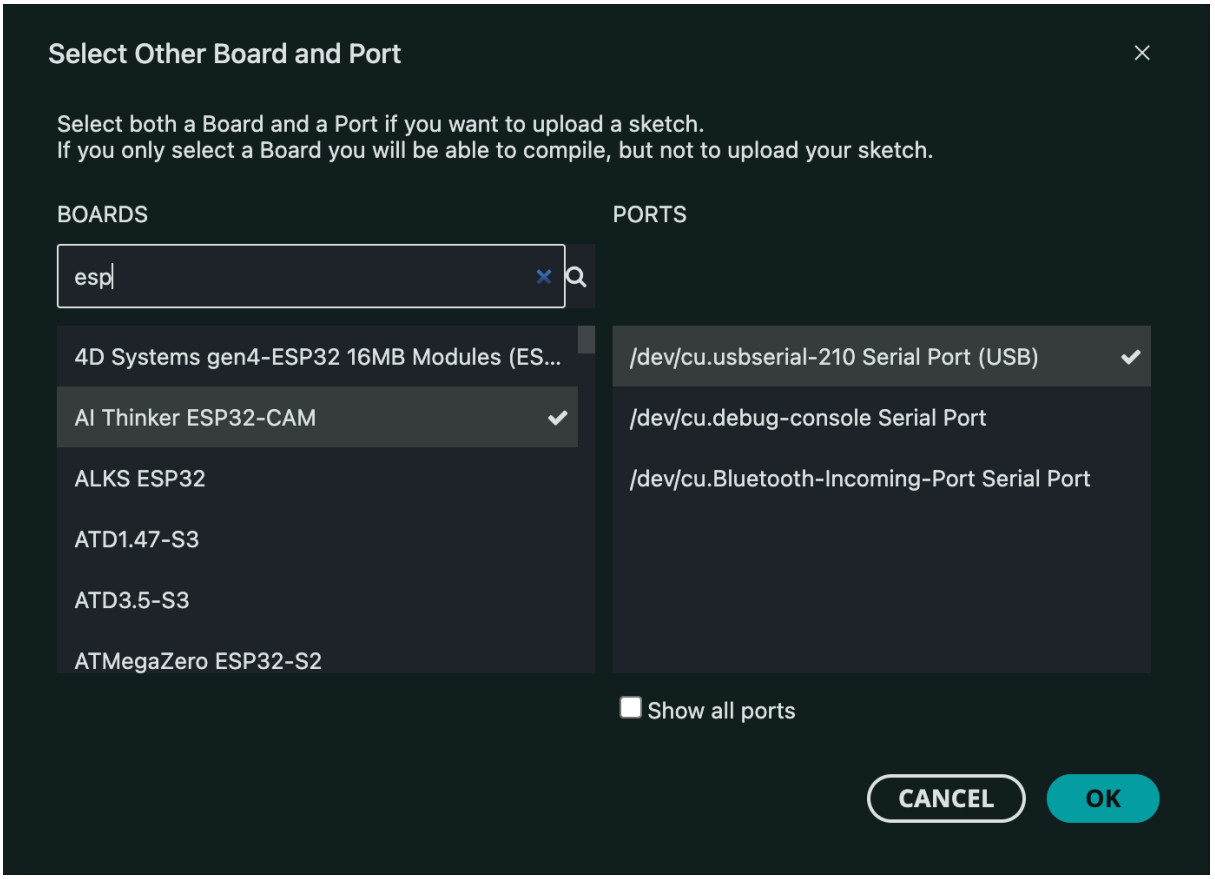
Download de Arduino IDE hier: https://www.arduino.cc/en/software/



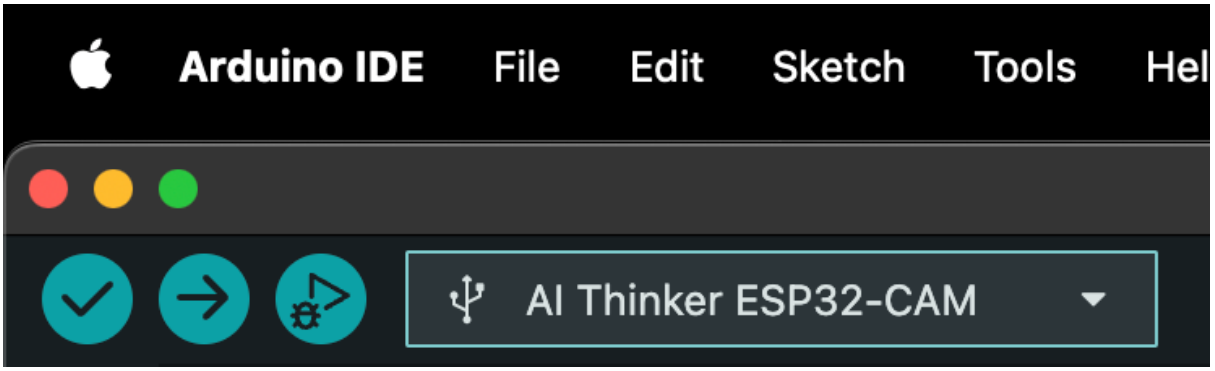
Na het installeeren van deze library kan je je board aansluiten en de juiste variant kiezen, in ons geval AI Thinker ESP32-CAM.



Druk op de Drop down en selecteer “Select Other Board and Port”



Selecteer hier vervolgens de “AI Thinker ESP32-CAM” en druk op OK. Nu weet de Arduino IDE welk Board er is aangesloten.

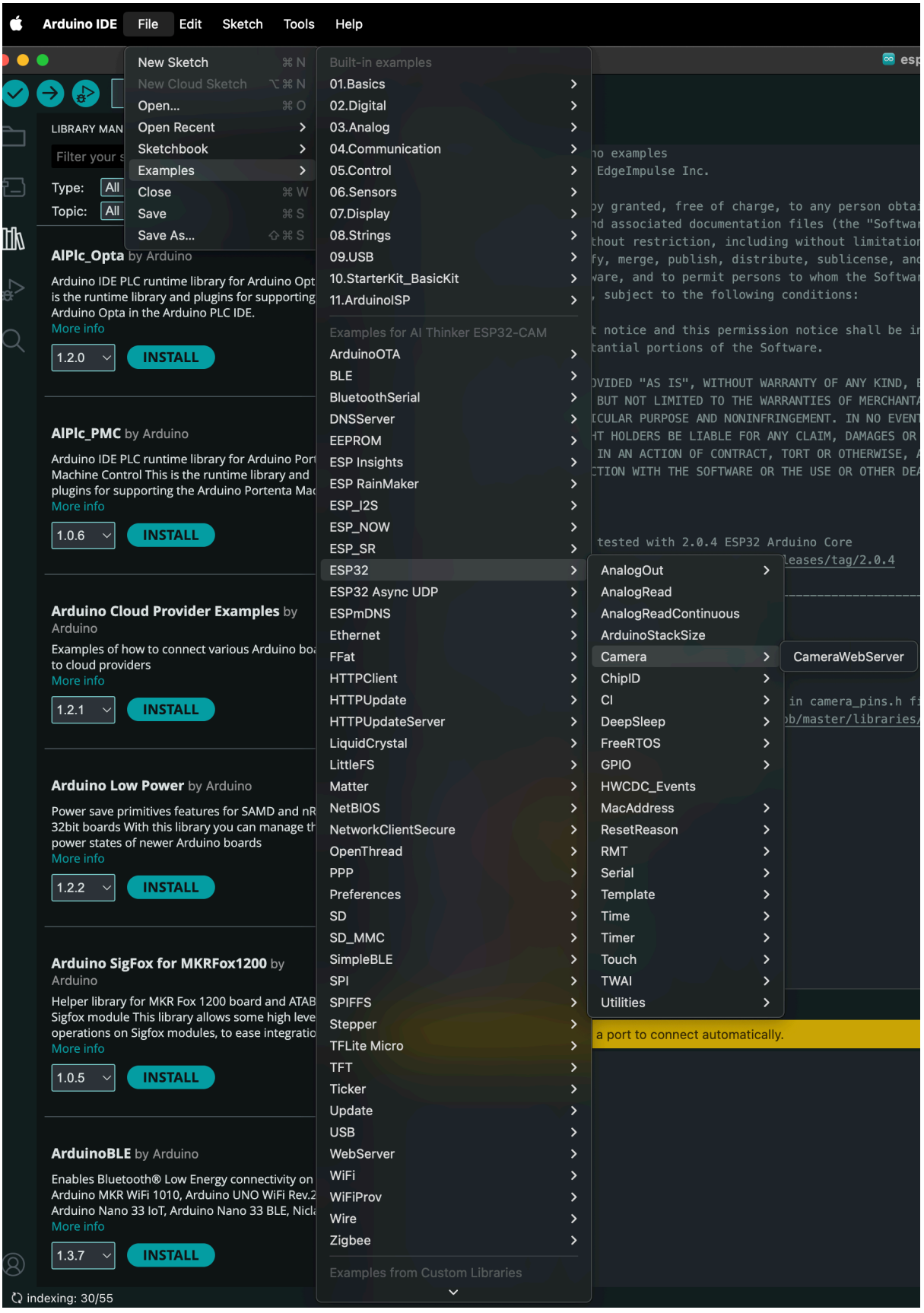


5. Data Collectie

Om een afbeelding classificatie model te trainen heb je natuurlijk fotos nodig!

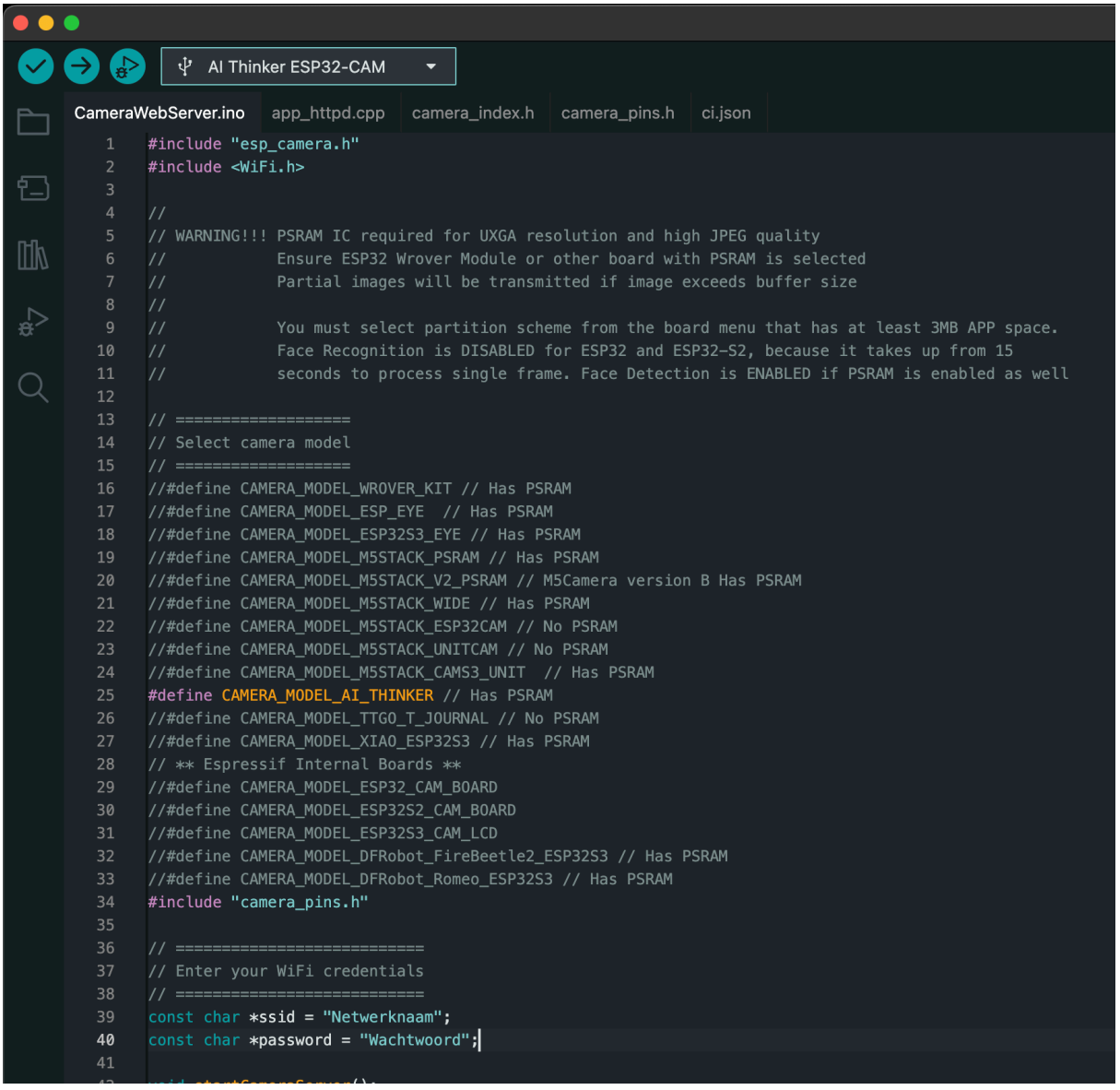
Deze gaan we verzamelen vanaf de ESP32-CAM zelf.

Dit doen we door gebruik te maken van de “CameraWebServer” Example die we in de vorige stap hebben geïnstalleerd via de “esp32” library.


Arduino IDE
File
Edit
Sketch
Tools
Help
New Sketch ⌘ N
New Cloud Sketch ⌥⌘ N
Open... ⌘ O
Open Recent
Sketchbook
Examples
Close ⌘ W
Save ⌘ S
Save As... ⇧⌘ S
Built-in examples
01.Basics
02.Digital
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB
10.StarterKit_BasicKit
11.ArduinoISP
Examples for AI Thinker ESP32-CAM
ArduinoOTA
BLE
BluetoothSerial
DNSServer
EEPROM
ESP Insights
ESP RainMaker
ESP_I2S
ESP_NOW
ESP_SR
ESP32
ESP32 Async UDP
ESPmDNS
Ethernet
FFat
HTTPClient
HTTPUpdate
HTTPUpdateServer
LiquidCrystal
LittleFS
Matter
NetBIOS
NetworkClientSecure
OpenThread
PPP
Preferences
SD
SD_MMC
SimpleBLE
SPI
SPIFFS
Stepper
TFLite Micro
TFT
Ticker
Update
USB
WebServer
WiFi
WiFiProv
Wire
Zigbee
Examples from Custom Libraries
AnalogOut
AnalogRead
AnalogReadContinuous
ArduinoStackSize
Camera
ChipID
CI
DeepSleep
FreeRTOS
GPIO
HWCDC_Events
MacAddress
ResetReason
RMT
Serial
Template
Time
Timer
Touch
TWAI
Utilities
CameraWebServer
AIPlc_Opta by Arduino
Arduino IDE PLC runtime library for Arduino Opt
is the runtime library and plugins for supporting
Arduino Opta in the Arduino PLC IDE.
More info
1.2.0
INSTALL
AIPlc_PMC by Arduino
Arduino IDE PLC runtime library for Arduino Por
Machine Control This is the runtime library and
plugins for supporting the Arduino Portenta Ma
More info
1.0.6
INSTALL
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino bo
to cloud providers
More info
1.2.1
INSTALL
Arduino Low Power by Arduino
Power save primitives features for SAMD and nR
32bit boards With this library you can manage th
power states of newer Arduino boards
More info
1.2.2
INSTALL
Arduino SigFox for MKRFox1200 by Arduino
Helper library for MKR Fox 1200 board and ATAB
Sigfox module This library allows some high leve
operations on Sigfox modules, to ease integratio
More info
1.0.5
INSTALL
ArduinoBLE by Arduino
Enables Bluetooth® Low Energy connectivity on
Arduino MKR WiFi 1010, Arduino UNO WiFi Rev.2
Arduino Nano 33 IoT, Arduino Nano 33 BLE, Nicl
More info
1.3.7
INSTALL
indexing: 30/55
a port to connect automatically.

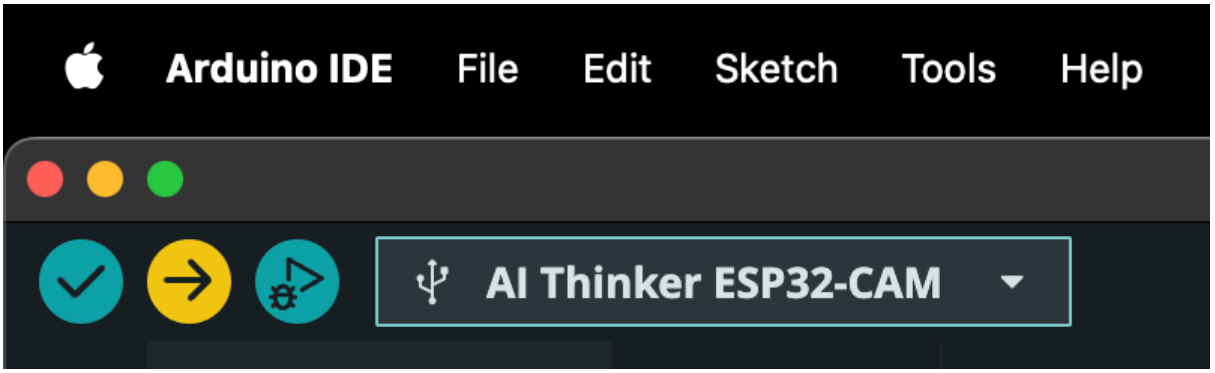
Na het openen van deze sketch moet je een paar instellingen aanpassen:

- Zorg ervoor dat het juiste model is geselecteerd zoals hieronder te zien. (`#define CAMERA_MODEL_AI_THINKER`)
- Vul het juiste `"ssid"` in. (LET OP: je hebt een 2,4GHZ netwerk nodig, het handigste is een eigen hotspot vanaf je telefoon wanneer je op school bent)
- Vul het correcte `"password"` voor dit netwerk in.

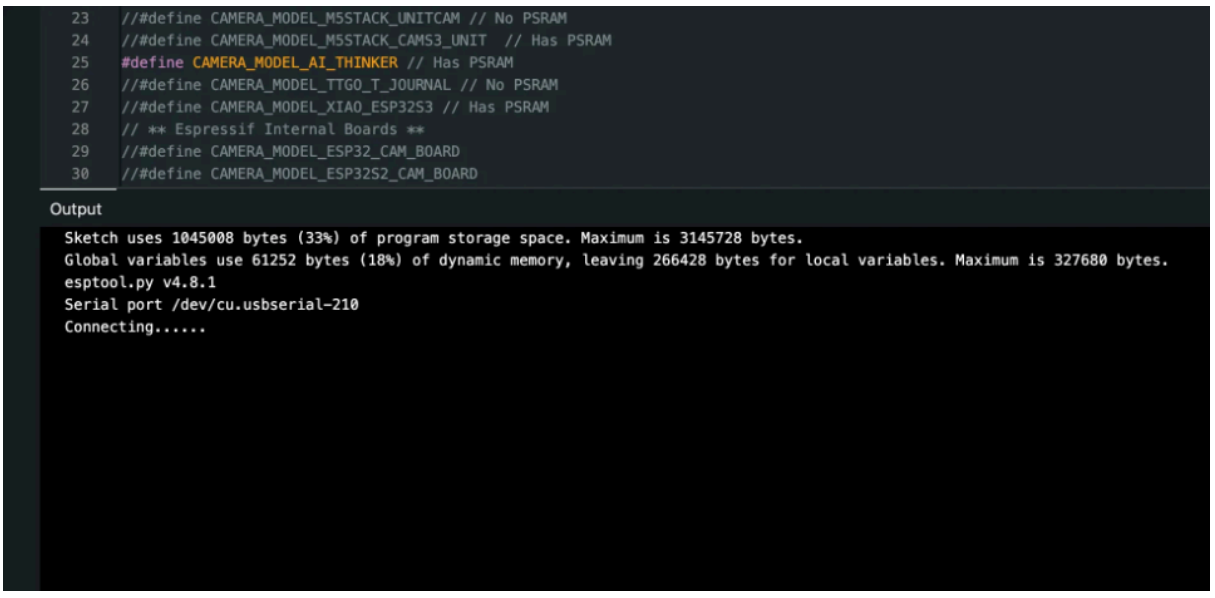


Na het aanpassen is het tijd om de sketch naar de ESP te flashen. Dit doe je door op de gele knop in de onderstaande afbeelding te drukken. LETOP:

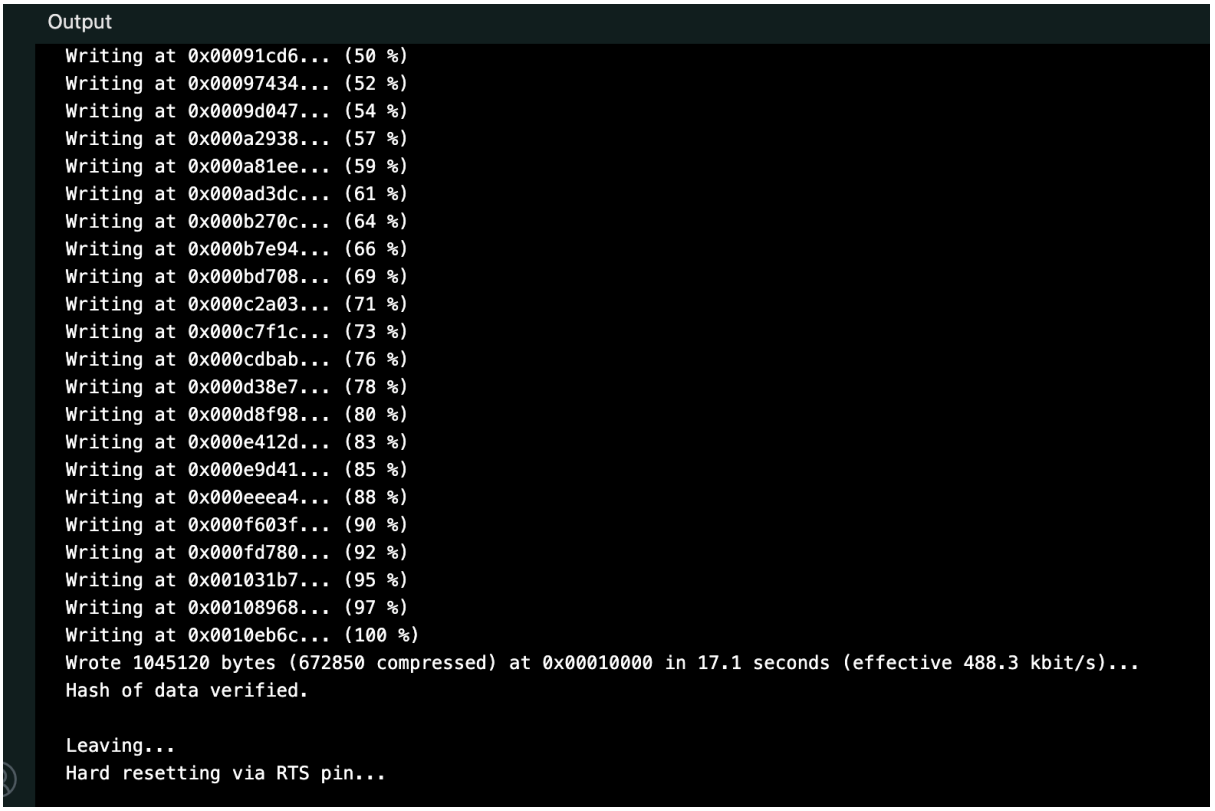
Selecteer het juiste bord in de dropdown (AI Thinker ESP32-CAM)



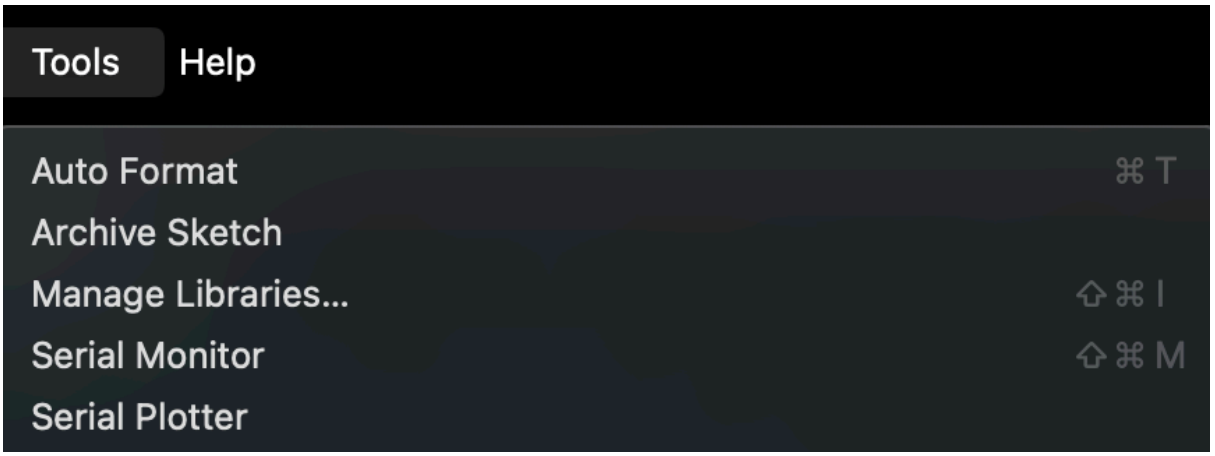
Nu zal het flash process beginnen, je Sketch wordt gecompiled en wachten op de connectie met de ESP. Wanneer “Connecting……” zichtbaar wordt moet je op de knopjes van de esp drukken, dit is soms een beetje friemelen. Wat het beste werkt is de RST knop op de achterkant van de esp 2 seconden in te drukken en daarna de twee zijknopjes in te drukken.



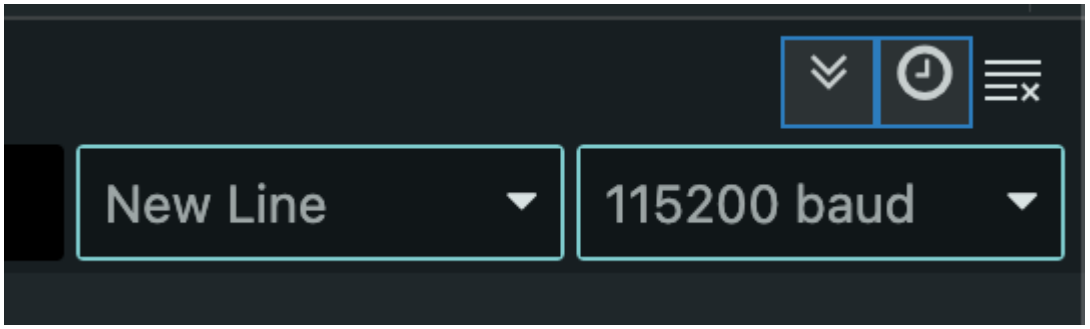
Wanneer de ESP is geconnect wordt het programma op de ESP geschreven een succesvolle flash ziet er als volgt uit



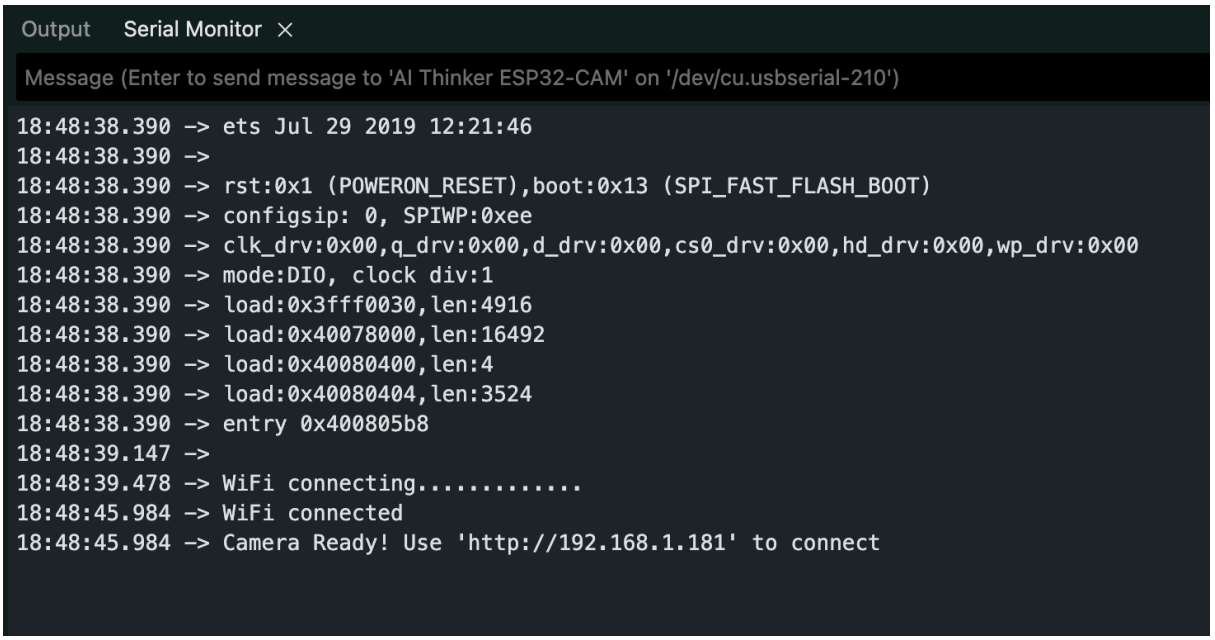
Druk nu op de RST knop zodat de ESP opnieuw opstart en ga naar de Serial Monitor



Zorg ervoor dat de Serial Monitor op 115200 baud staat, als je gekke tekens in de Serial Monitor ziet dan is dit waarschijnlijk het probleem.

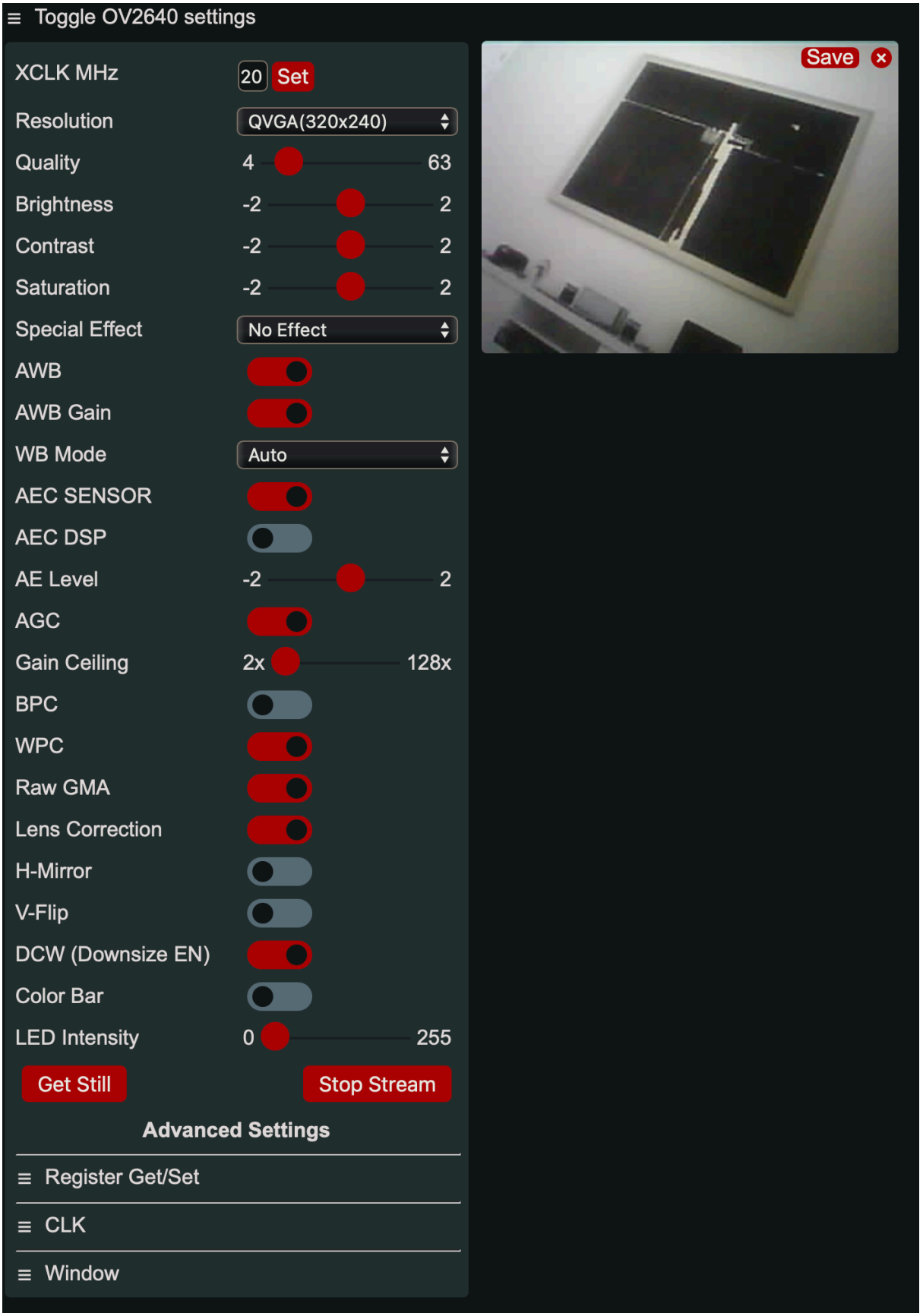


Als alles goed is gegaan zie je nu een ip address in de Serial Monitor staan.



Kopieer de http link in je browser om naar de CameraWebServer te gaan. LETOP je moet op hetzelfde netwerk zitten als dat de ESP-32 Zit (de ingevulde SSID).

Hier zul je het onderstaande menu aantreffen, druk op start stream om een live Feed te bekijken.


Toggle OV2640 settings
XCLK MHz
20
Set
Resolution
QVGA(320x240)
Quality
4
63
Brightness
-2
2
Contrast
-2
2
Saturation
-2
2
Special Effect
No Effect
AWB
AWB Gain
WB Mode
Auto
AEC SENSOR
AEC DSP
AE Level
-2
2
AGC
Gain Ceiling
2x
128x
BPC
WPC
Raw GMA
Lens Correction
H-Mirror
V-Flip
DCW (Downsize EN)
Color Bar
LED Intensity
0
255
Get Still
Stop Stream
Advanced Settings
Register Get/Set
CLK
Window
Save

Hier kan je verschillende instellingen aanpassen zoals de resolutie, het is handig om de hoogst mogelijke resolutie te kiezen, deze kan je later altijd nog aanpassen. Beter teveel dan te weinig.

Nu kan je een Dataset gaan maken!

EdgeImpulse adviseert om minstens 20 foto's per klasse te maken, ook hier geld beter te veel dan te weinig. Let wel op de balans van je klassen, in het optimale geval heb je evenveel fotos van iedere klasse die je hebt.

Het is handig om een map aan te maken voor de verschillende klasse die je hebt, hierdoor kan je later in edge impulse snel je data van een Label voorzien.

6. EDGE Impulse

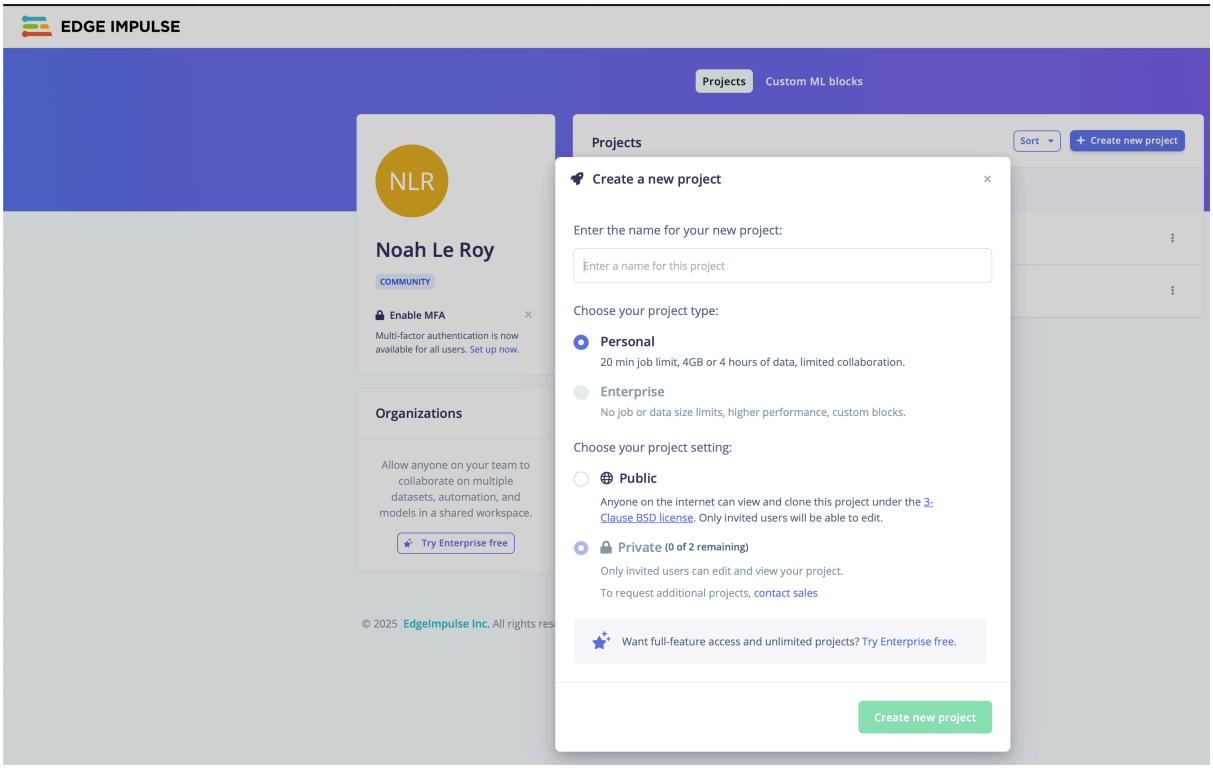
Edge Impulse is een platform waarmee je machine learning-modellen kunt ontwikkelen, trainen en inzetten op edge devices zoals microcontrollers en sensoren. Het richt zich op toepassingen in embedded AI, zoals spraakherkenning, beeldanalyse of trillingsdetectie. Wij gaan het gebruiken om een simpel image classification model mee te maken.

6.1 Project Setup

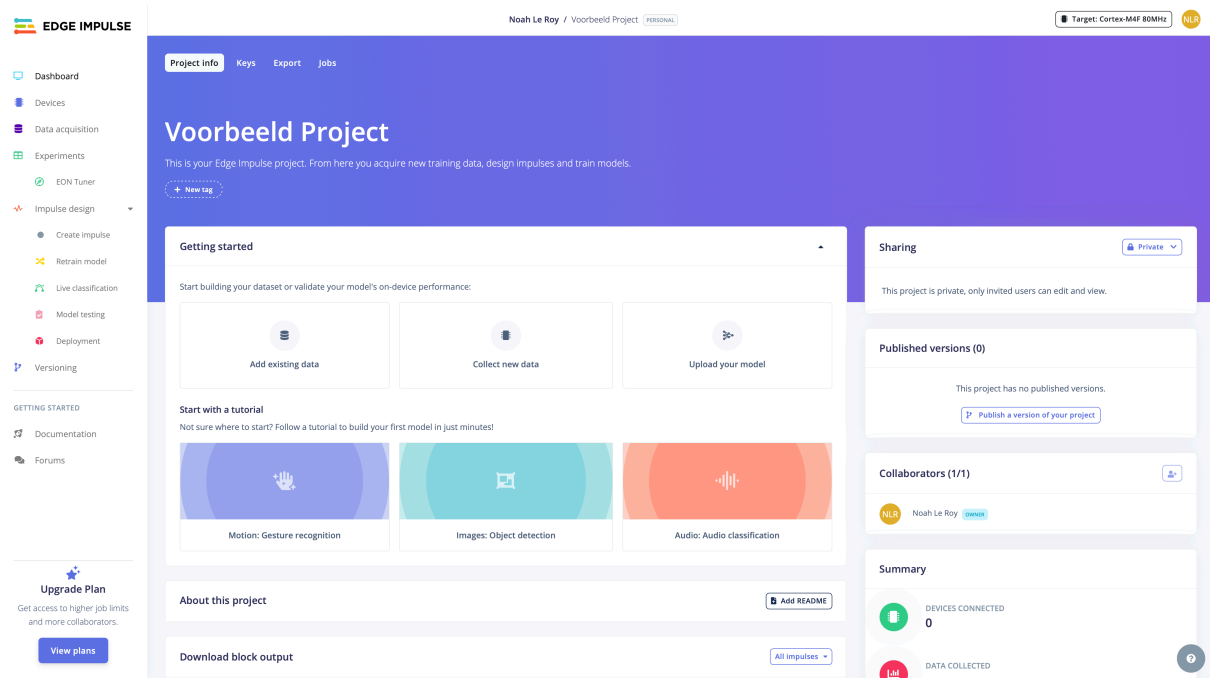
Maak een Edge Impulse account aan op edgeimpulse.com

wanner je bent ingelogd zal je het onderstaande zien, druk hier op "Create new project" en geef

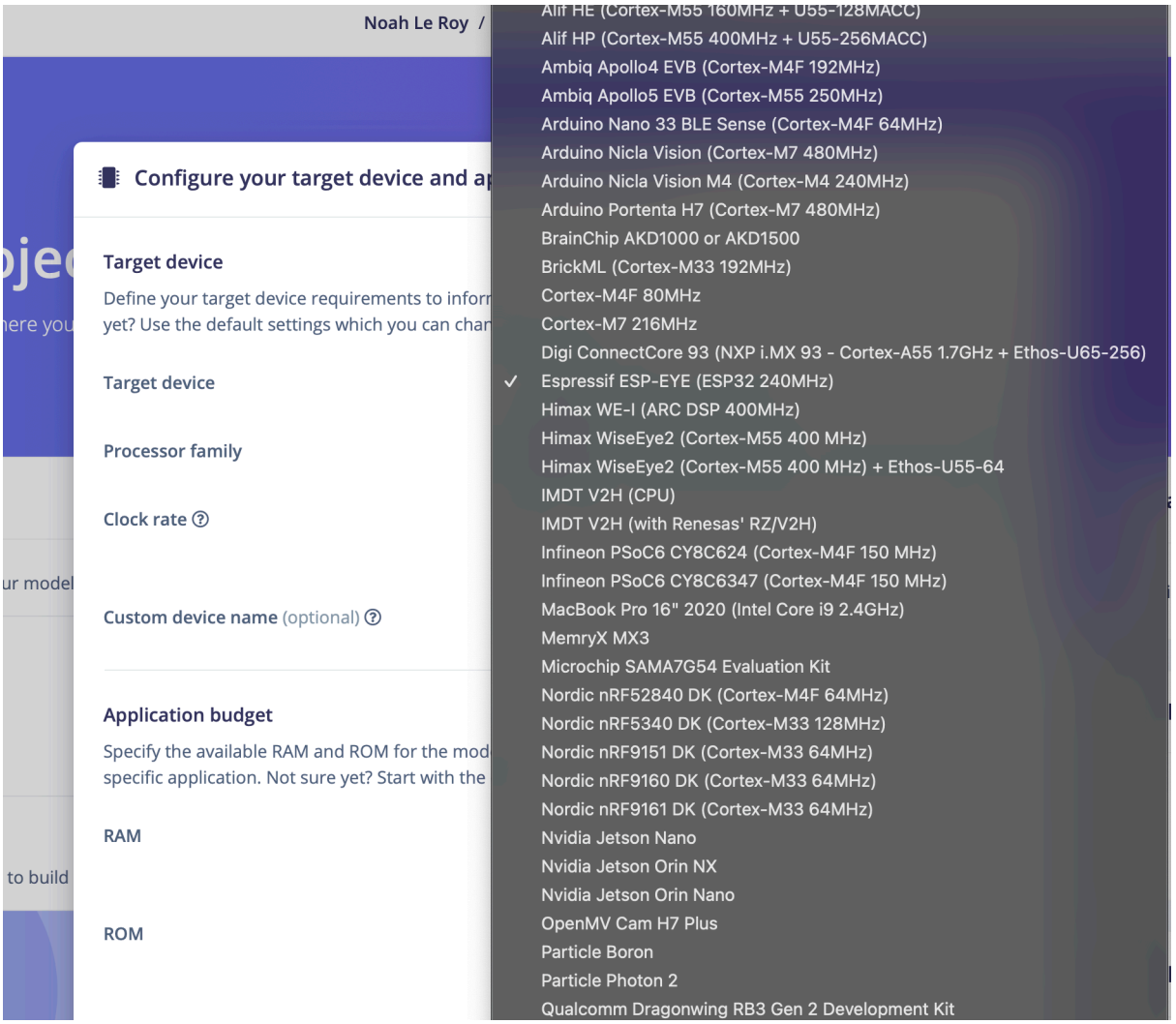
het project een naam.



Na het aanmaken van het project kom je op dit scherm.



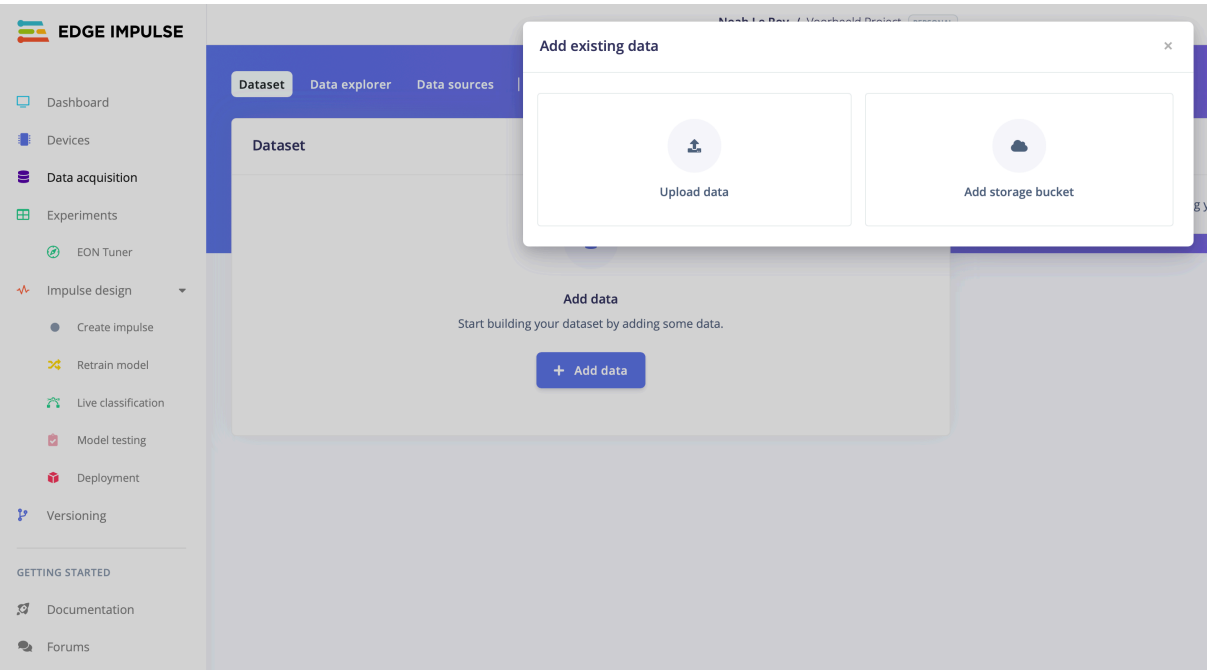
Rechts Boven aan zie je “Target” staan. druk hier op en selecteer de ESP32-EYE en druk op “Save”. Dit is niet exact onze ESP, maar Close enough!



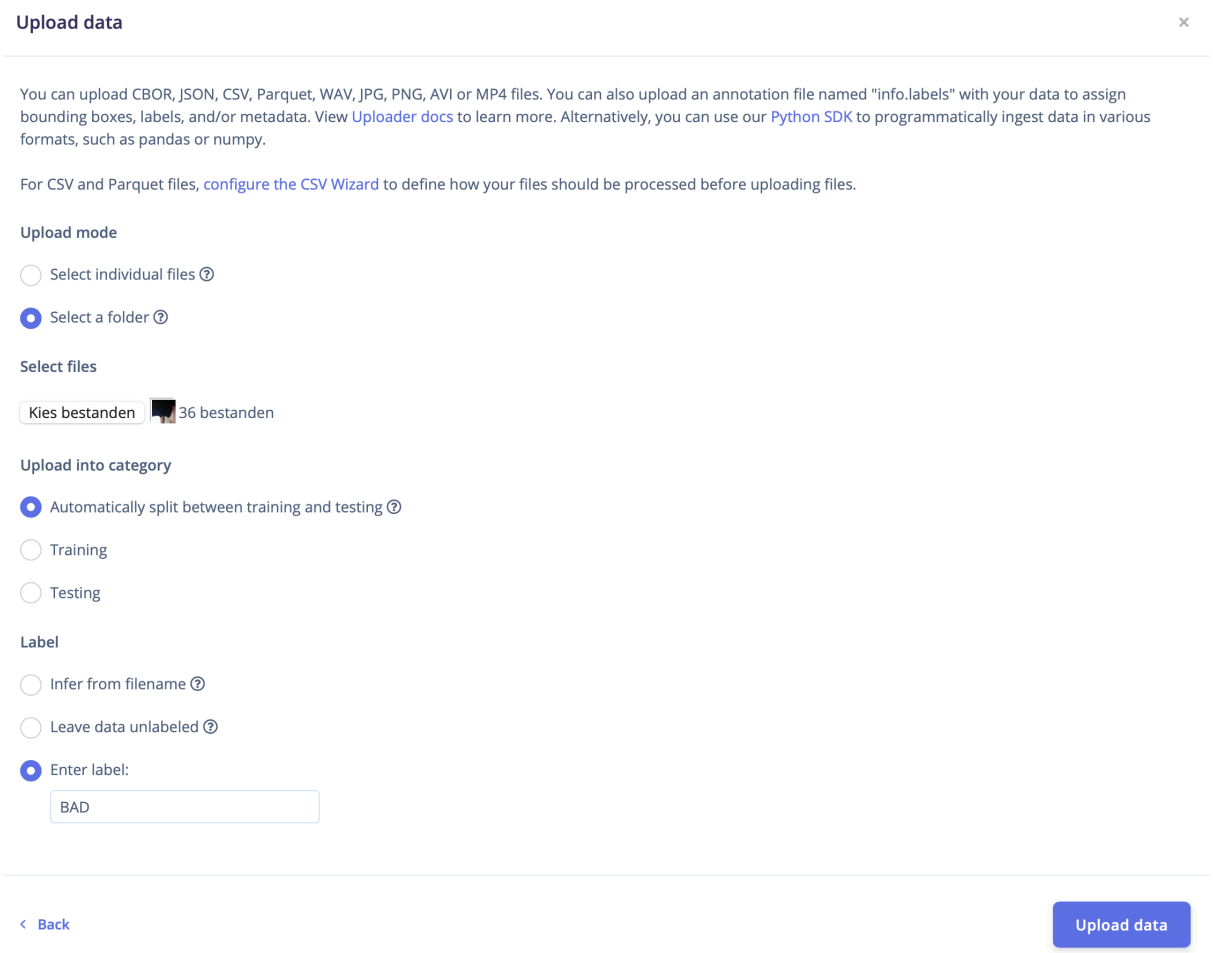
6.2 Data Uploaden

Nu zijn we klaar om onze Dataset te uploaden.

Ga naar “Data acquisition” aan de linker kant en druk vervolgens op “Add data”

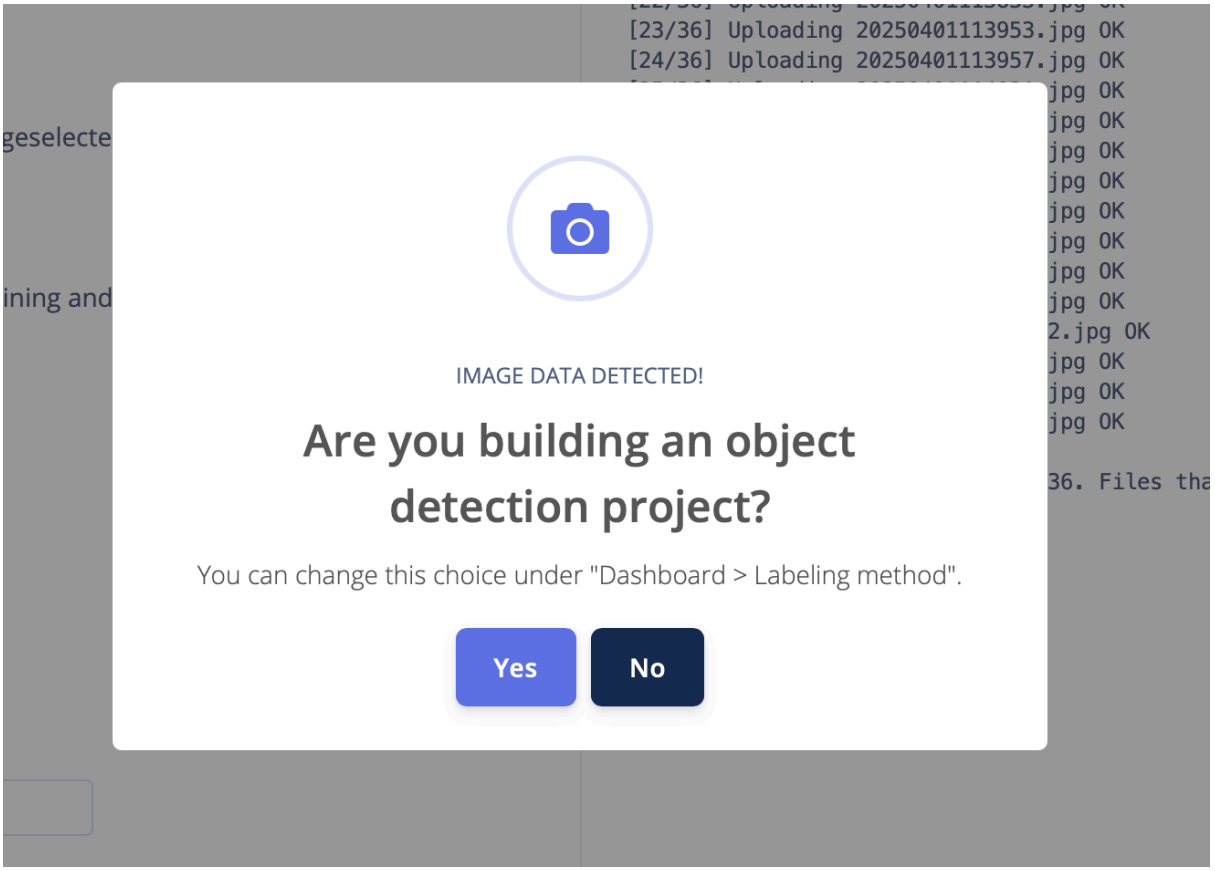


Druk op “Upload data”. Hier kan je per klasse een map uploaden en meteen van een Label voorzien zoals hieronder weergegeven.



Je kan hier zelf aangeven hoe je wil dat de Split van de data gebeurd, Automatisch is prima tenzij je zelf een Train en Test set hebt gemaakt.

Druk op “Upload data”



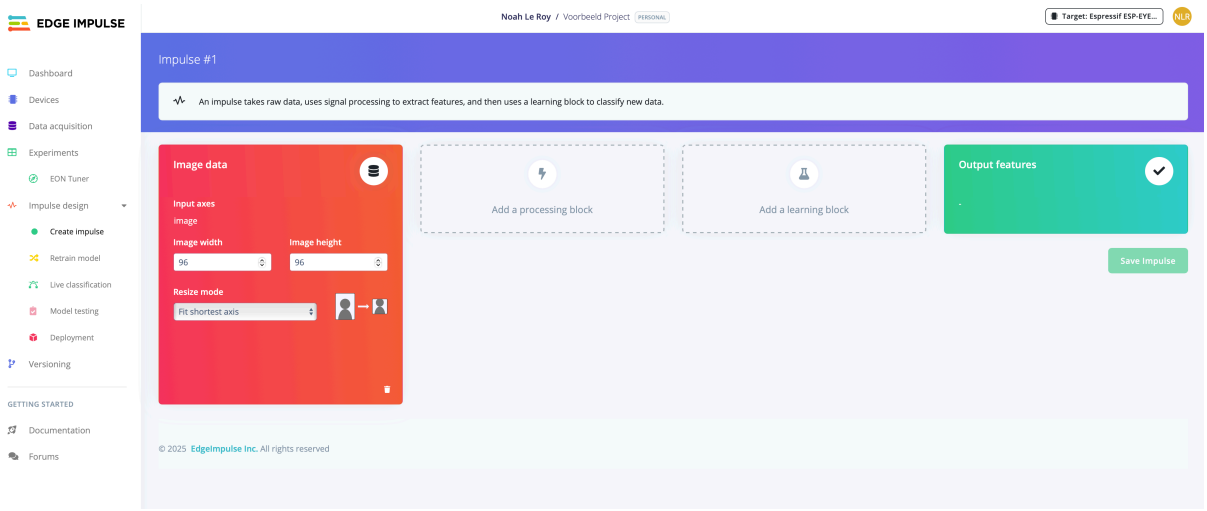
Selecteer hier “No” wij maken een image classification model, geen object detection model.

Upload nu op dezelfde manier je andere classes, Let op dat je ze het juiste Label toe kent.

6.3 Impulse Design

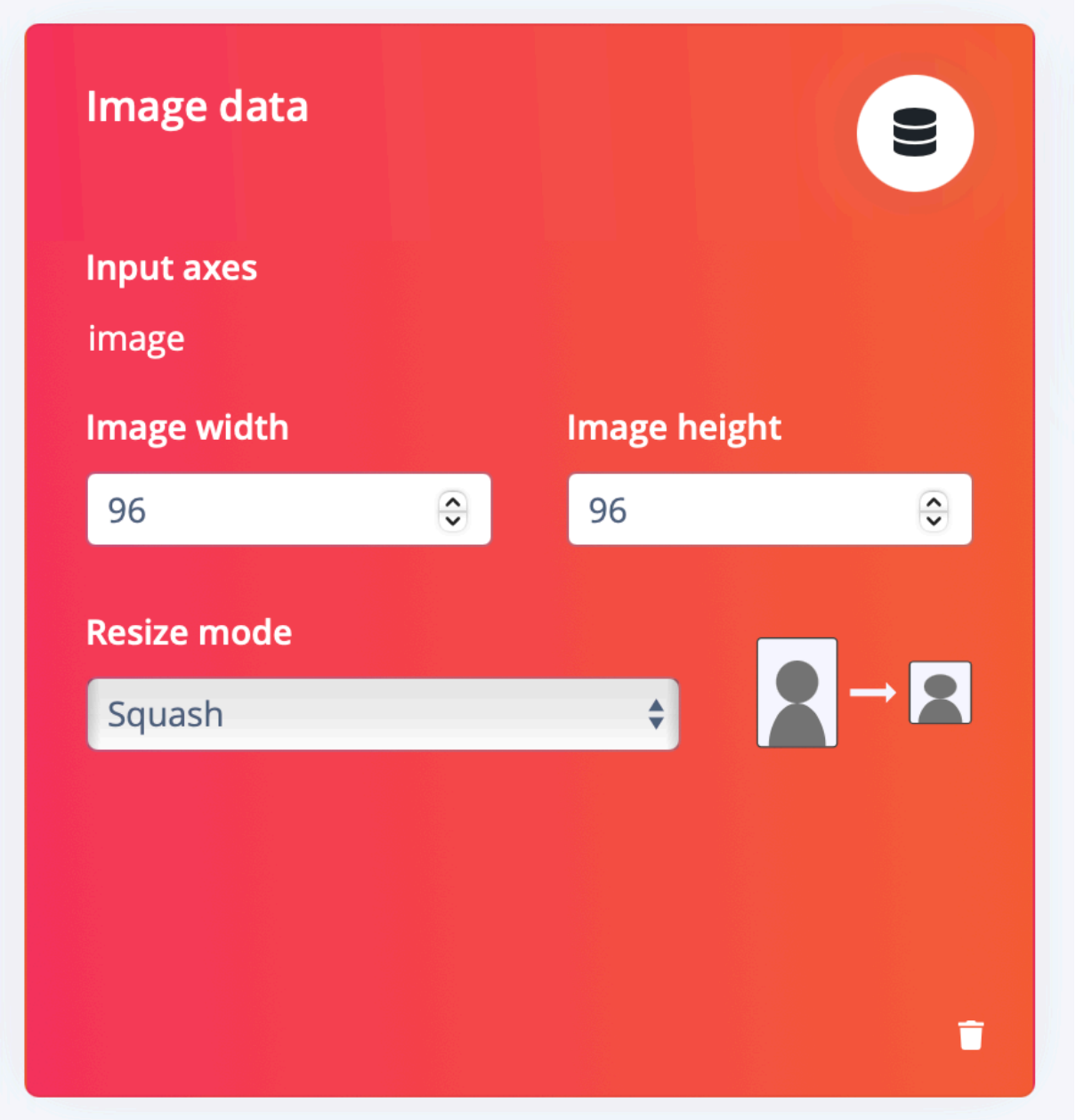
Nu de data is geüpload kunnen we aan het model beginnen.

ga naar “Impulse design” aan de linker kant.

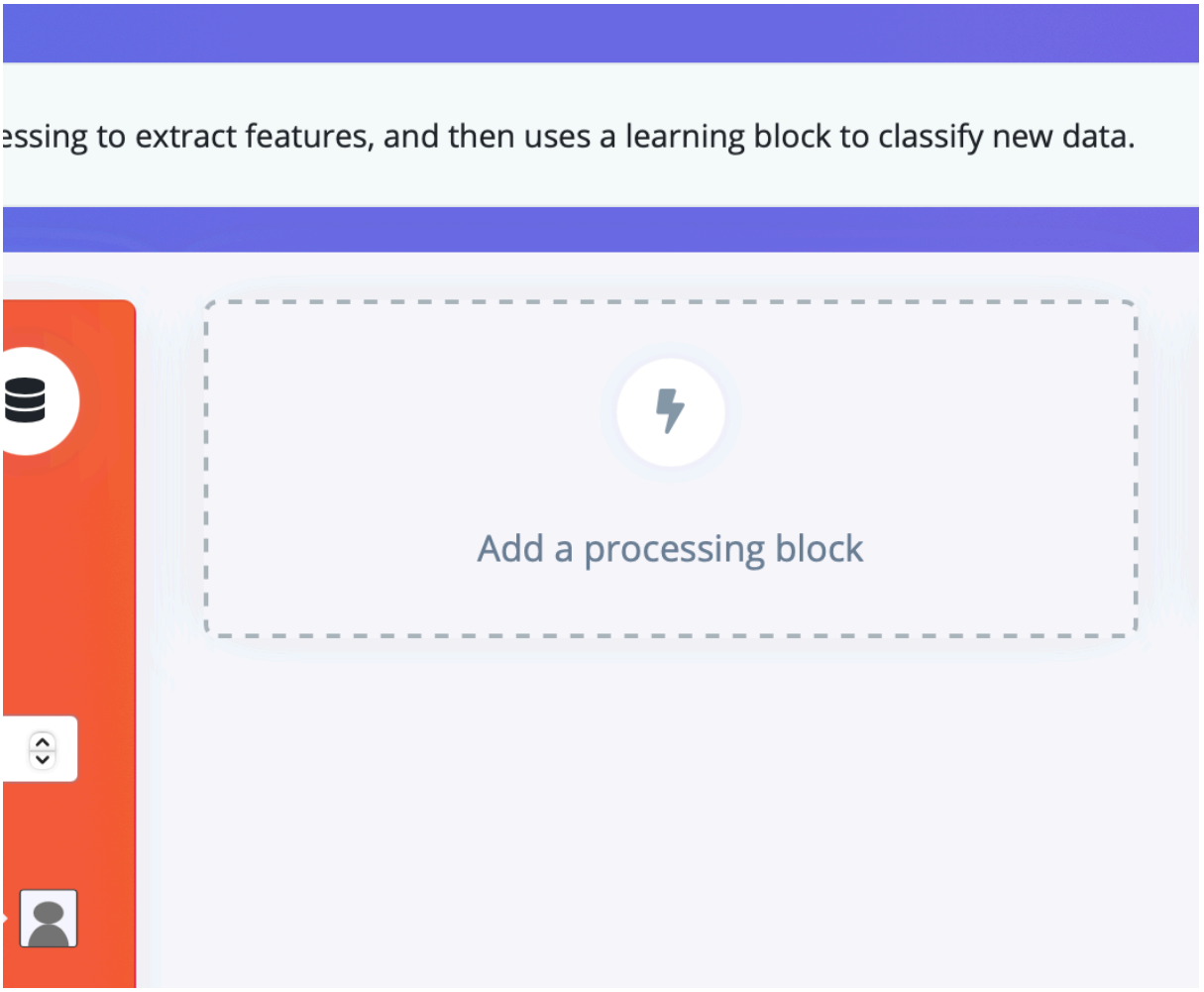


Hier gaan we ons model opzetten. We werken van links naar rechts.

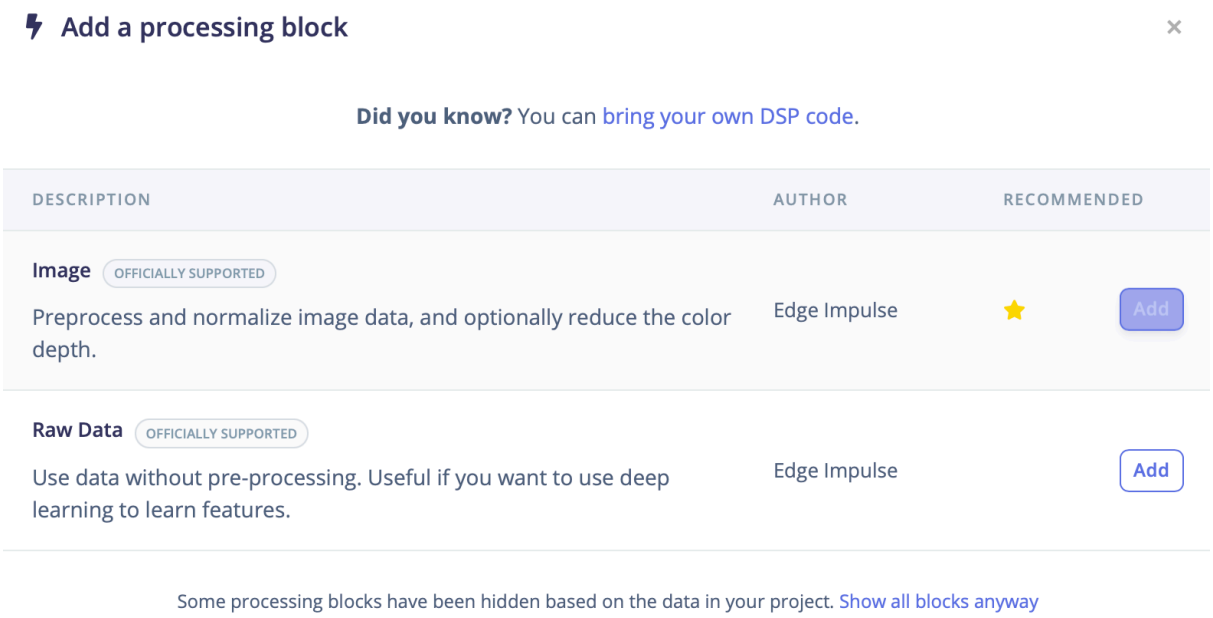
Eerst resizen we de afbeelding voordat deze het model in gaat, dit zorgt ervoor dat het model sneller wordt. Hoe groter de afbeelding, hoe langer het model er over doet om een nieuwe afbeelding te classificeren. ook drukken we de afbeelding samen (Squash) om zo veel mogelijk informatie te behouden als de afbeelding niet overeenkomt met de opgegeven verhouding.



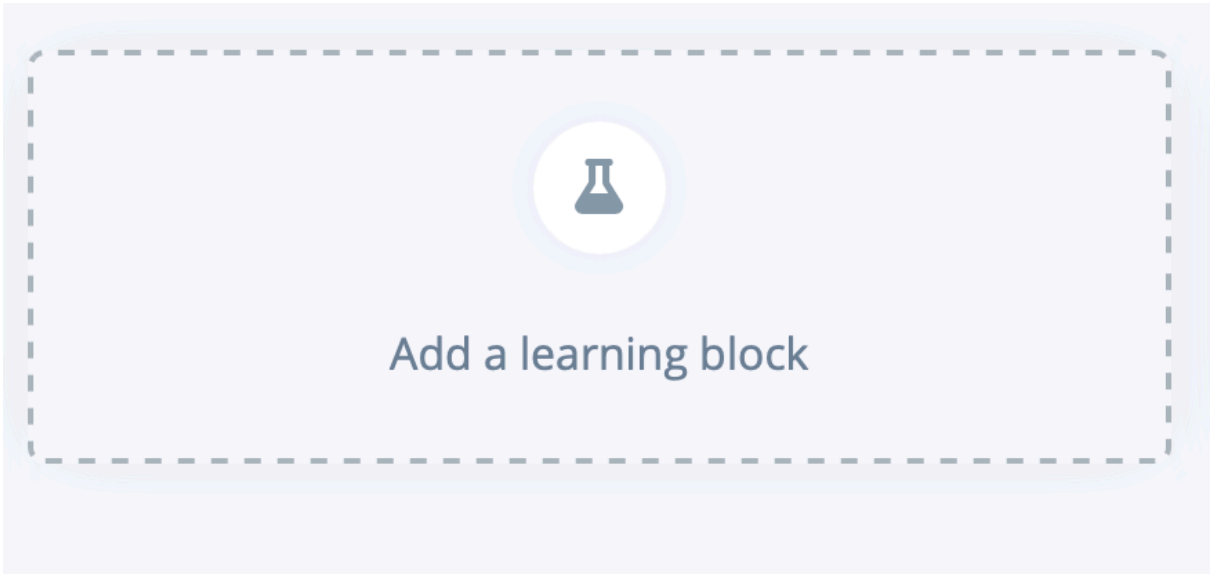
Druk nu op “Add a processing block”



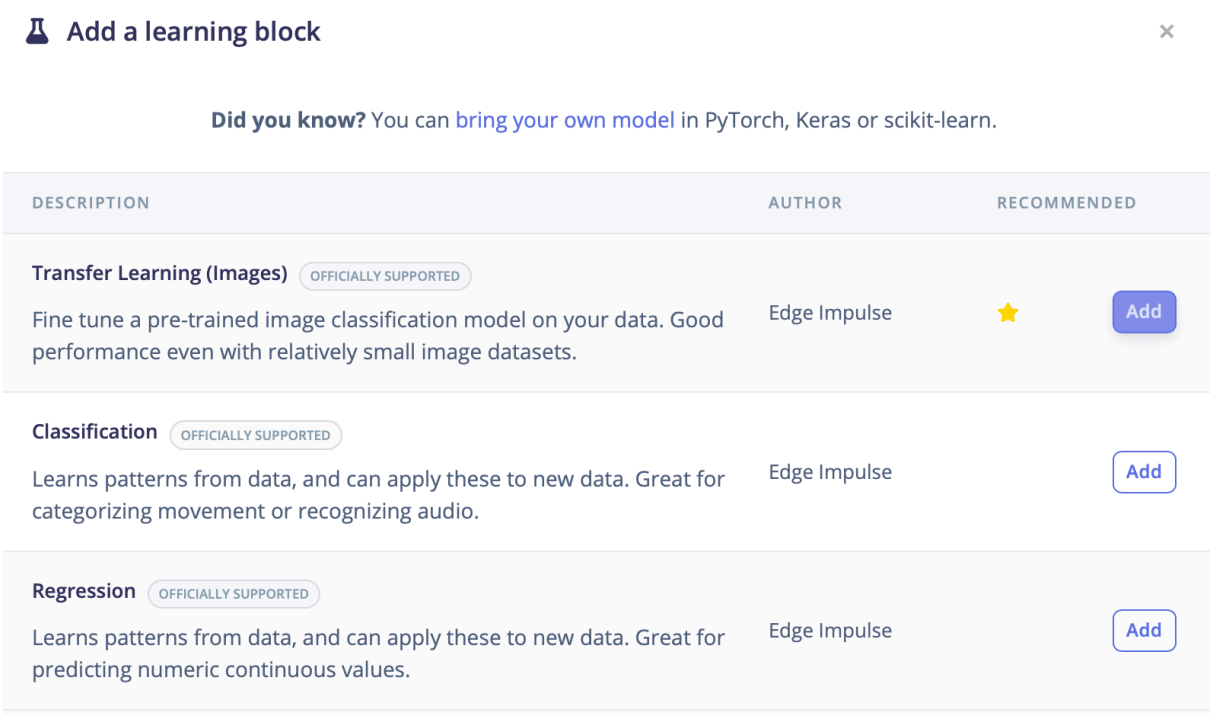
Selecteer hier “Image”



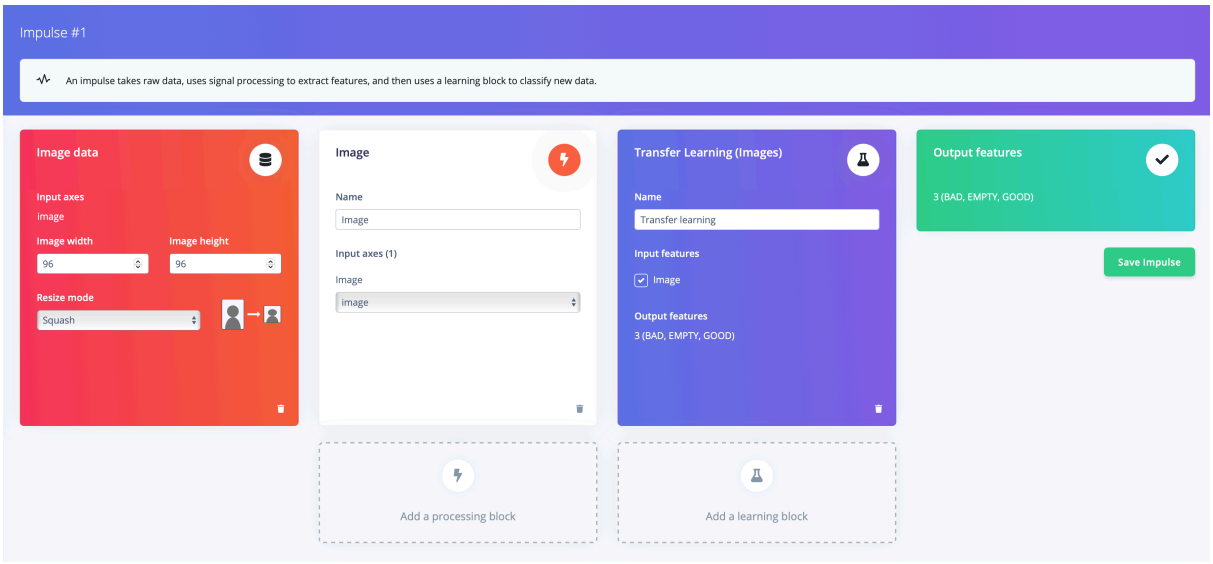
Druk nu op “Add a learning block”



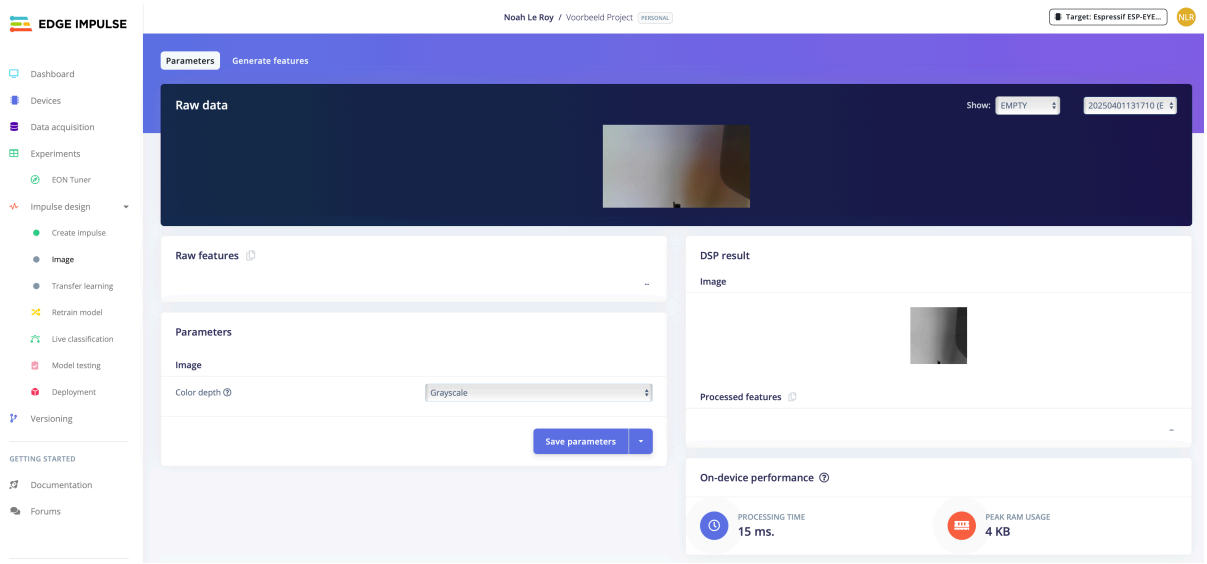
Selecteer hier “Transfer learning (Images)”



Je Impulse is nu klaar en ziet er als het goed is zo uit

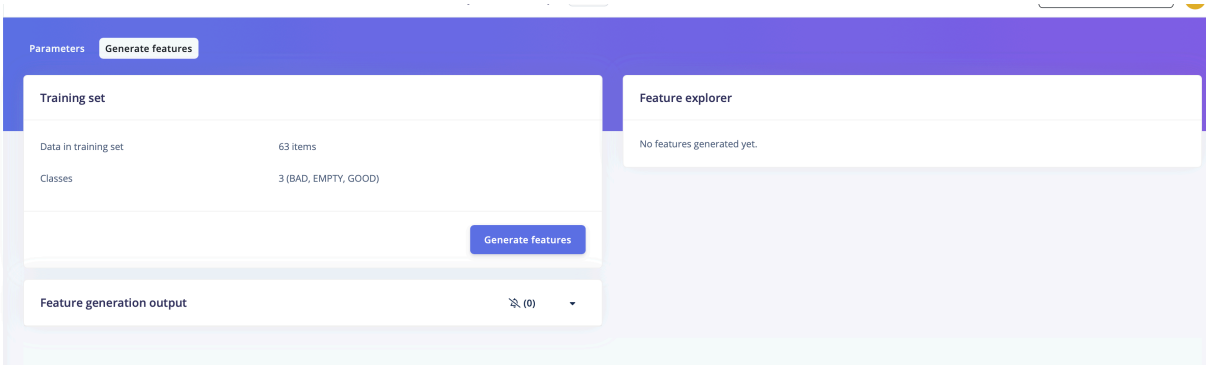


Druk nu op “Image” aan de linker kant onder “Create impulse”



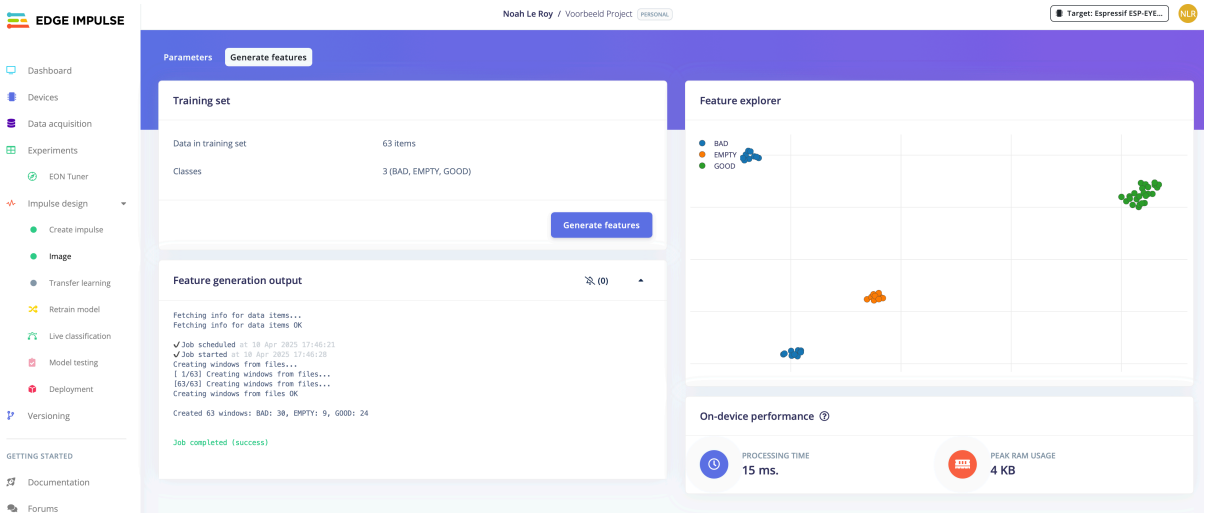
Hier kunnen we de colorspace aanpassen. “Grayscale” zal sneller werken op de ESP maar als kleur een belangrijk onderdeel is van je classificatie kan je hem op “RGB” zetten.

Druk vervolgens op “Save parameters”



Je wordt doorgestuurd naar dit scherm, hier ga je features genereren. Check of al je classes er tussen staan. in dit geval “GOOD”, “BAD” en “EMPTY”.

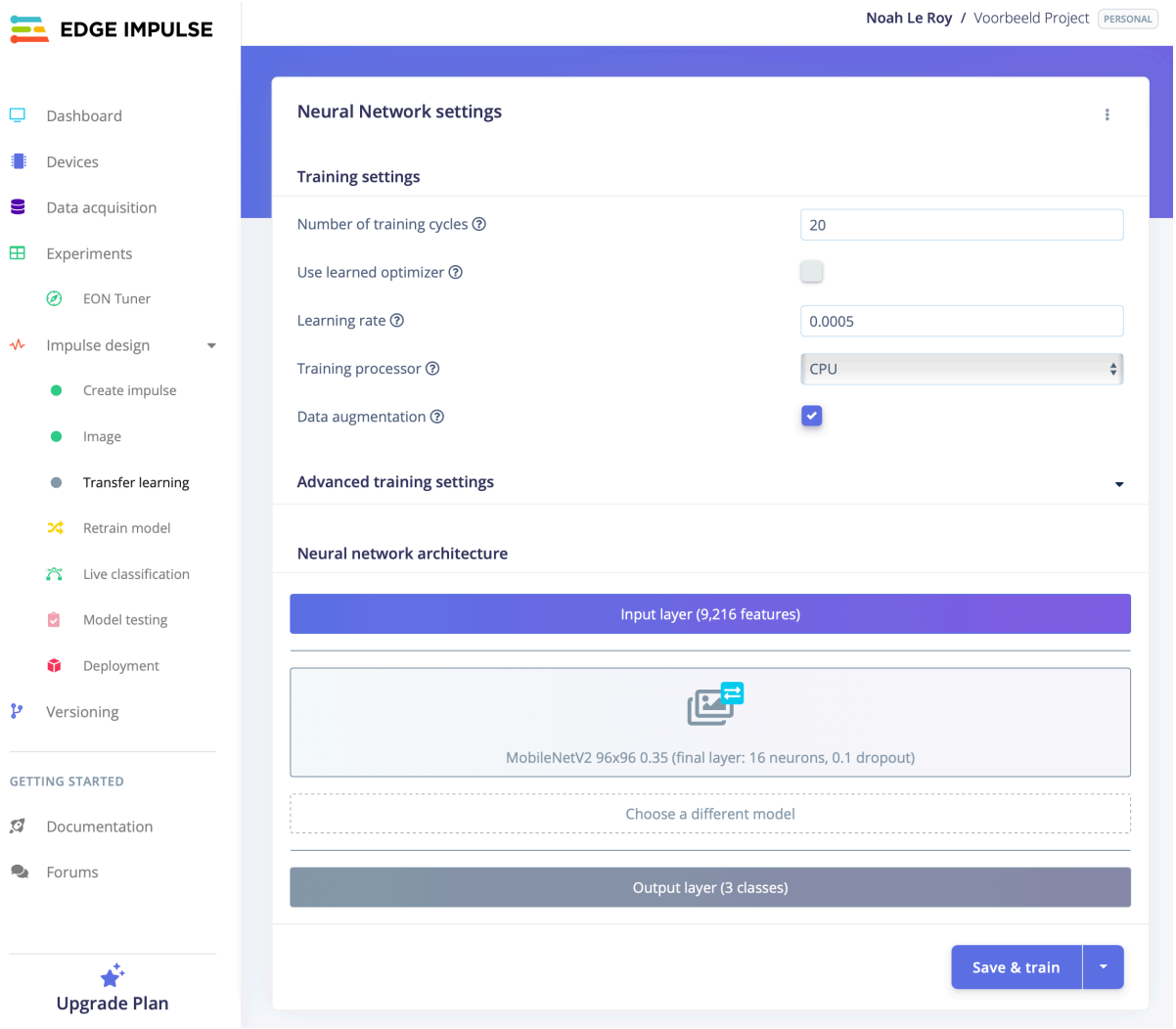
Druk op “Generate features”



Na het genereren van de Features zie je als het goed is een aantal clusters aan de rechterkant. Dit is goed!

Klik nu op “Transfer learning” aan de linker kant onder “Image”.

6.4 Model Trainen

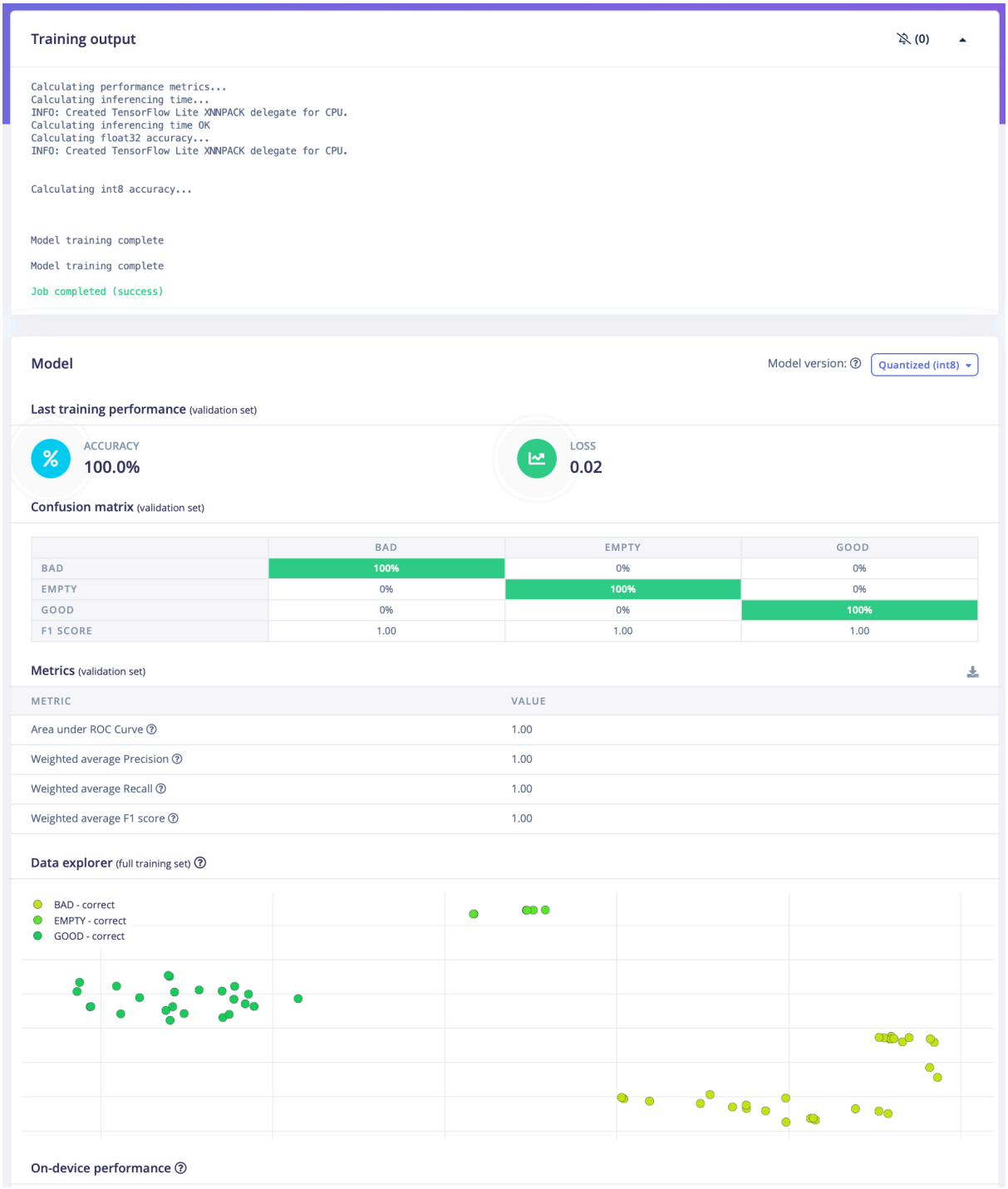


In dit menu gaan we het model trainen. Je kan gewoon beginnen met de standaard instellingen, als je ziet dat het model niet genoeg leert kan je altijd proberen te spelen met de “Number of training cycles” en de “Learning rate”

Het is handig om “Data augmentation” te selecteren, dit zal je images over een aantal assen spiegelen zodat het model meer data heeft om van te leren.

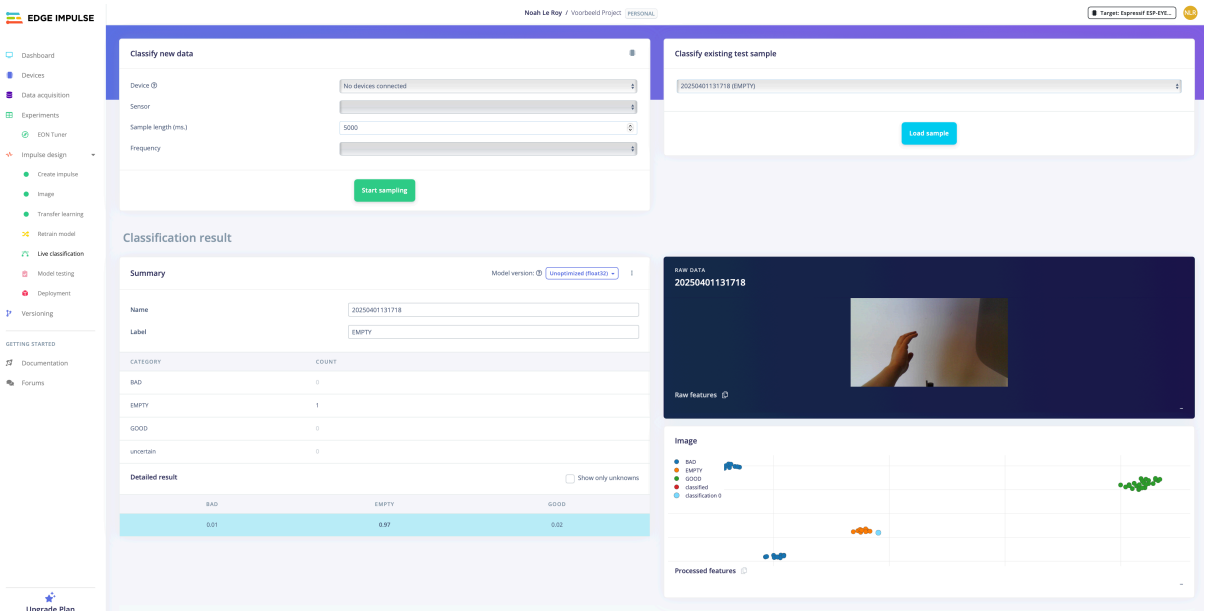
Druk op “Save & train”

Het model wordt nu getraind, wacht rustig af totdat de output zegt dat het klaar is.

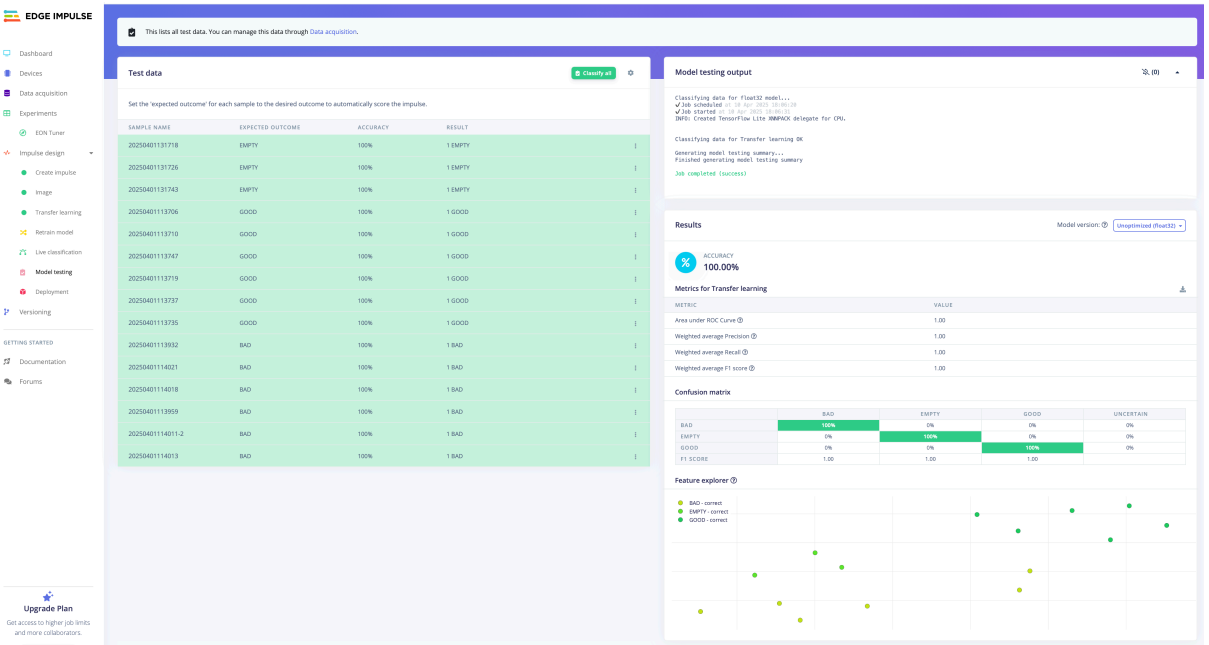


Hier zie je de performance op de validatie set, zoals te zien scoort dit model erg goed omdat er een duidelijk verschil in de afbeeldingen zit. Aan de clusters is te zien dat het model goed in staat is om de verschillende classes van elkaar te scheiden.

Ga nu naar “Live clasification” aan de linker kant onder “Retrain model”



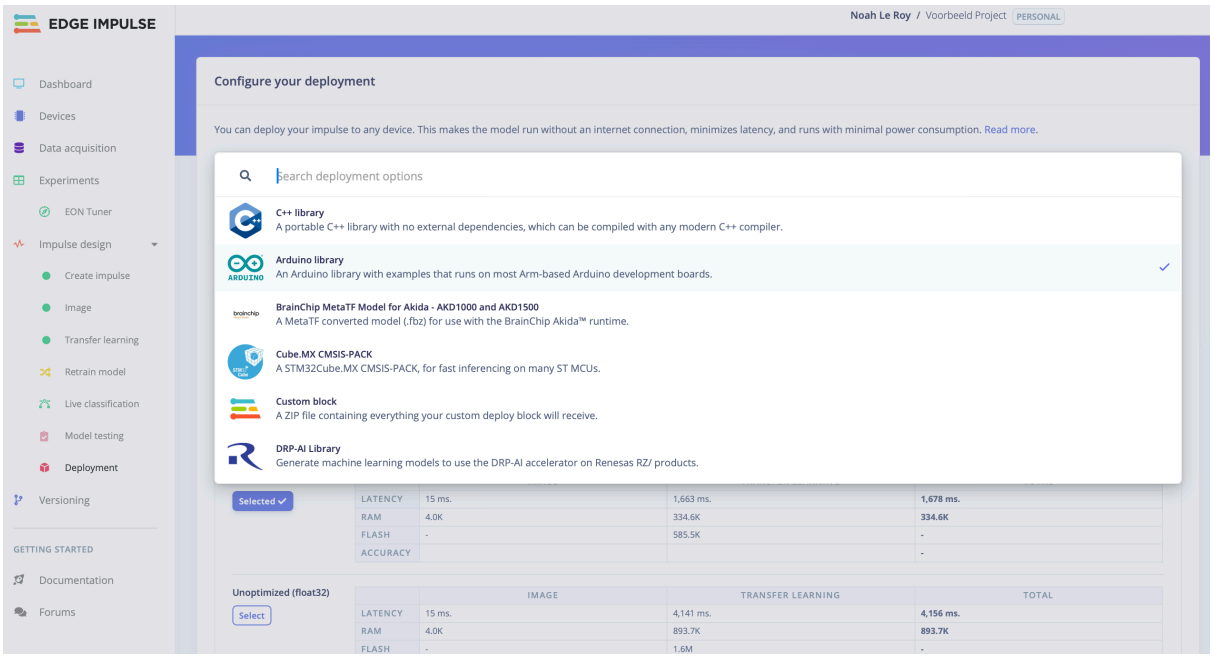
Hier kan je een foto selecteren uit je dataset en kijken of deze goed geclassificeerd wordt. Je kan ook naar “Model testing” gaan om het model los te laten op de gehele test set en de resultaten te bekijken. De Test set is aangemaakt tijdens het uploaden van de data (Train Test Split).



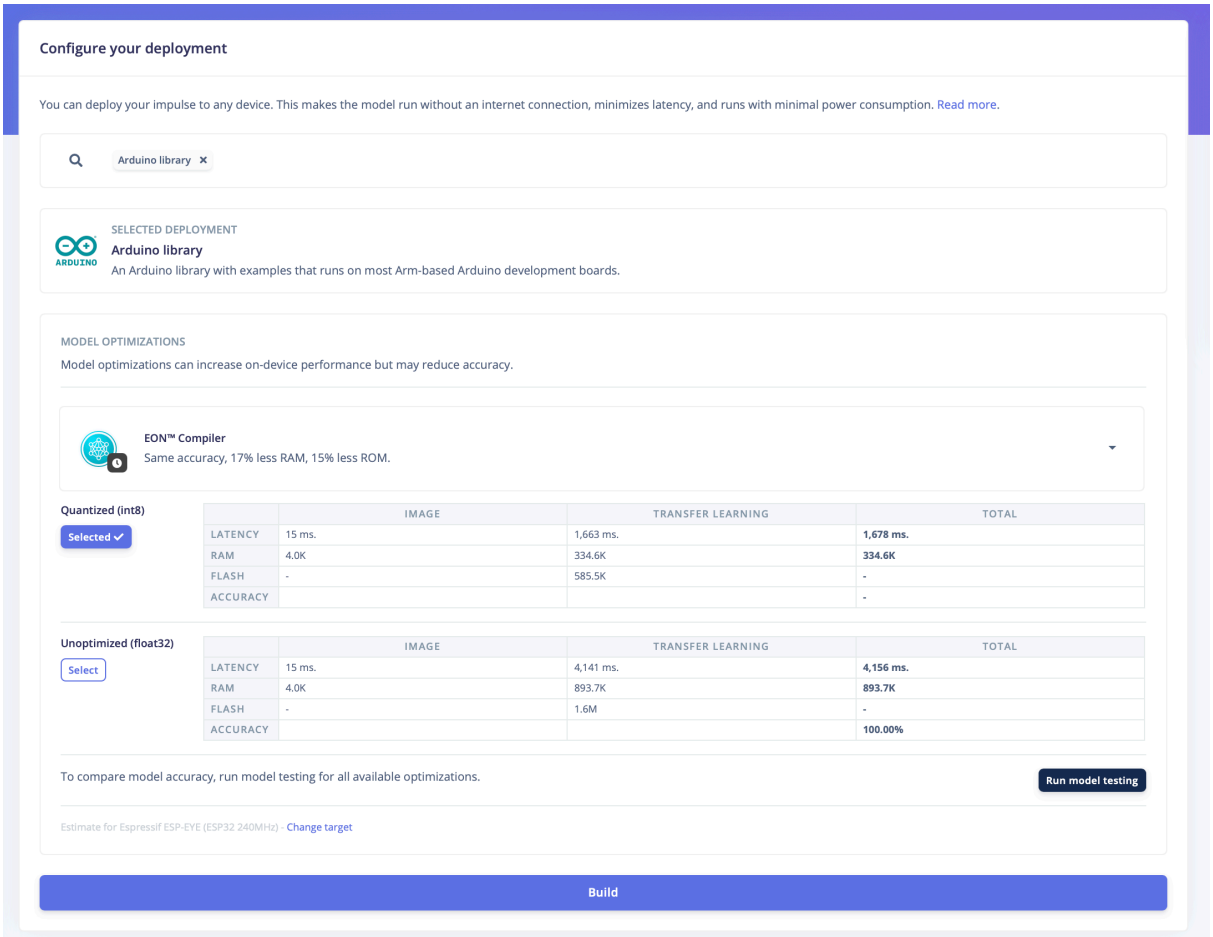
Als je tevreden bent met het resultaat is het tijd om het model te deployen op de ESP!

Klik op “Deployment” aan de linker kant onder “Model Testing”

7. Deployment

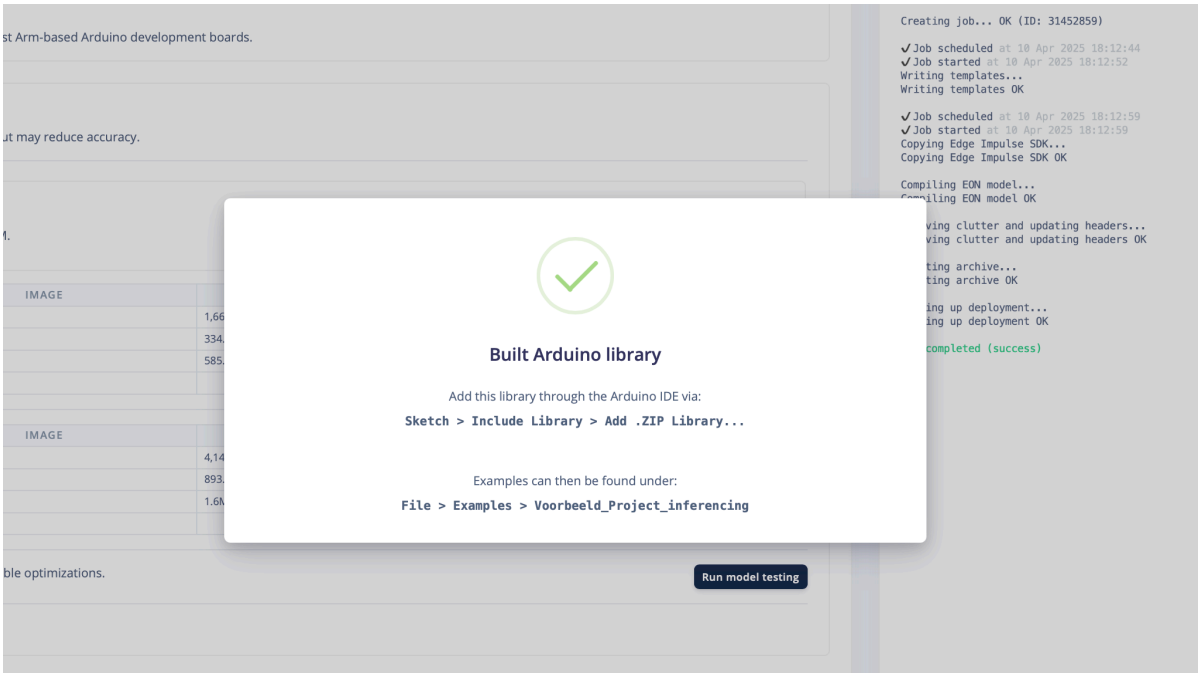


Druk op de zoekbalk en selecteer “Arduino library”

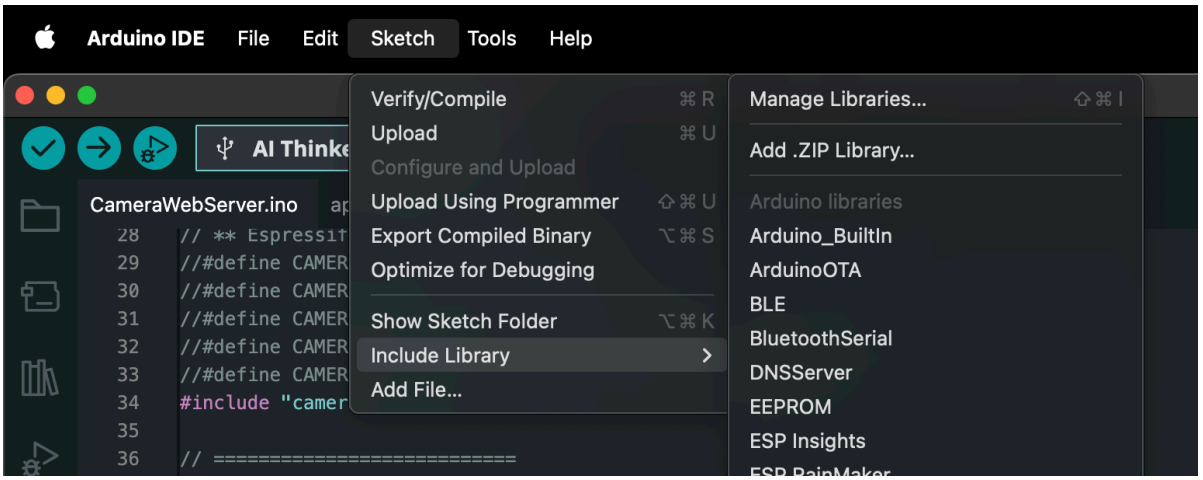


Hier kan je kiezen of je het model wil Quantizen of niet. Een Quantized model zal sneller draaien op de ESP maar mogelijk minder accuraat zijn.

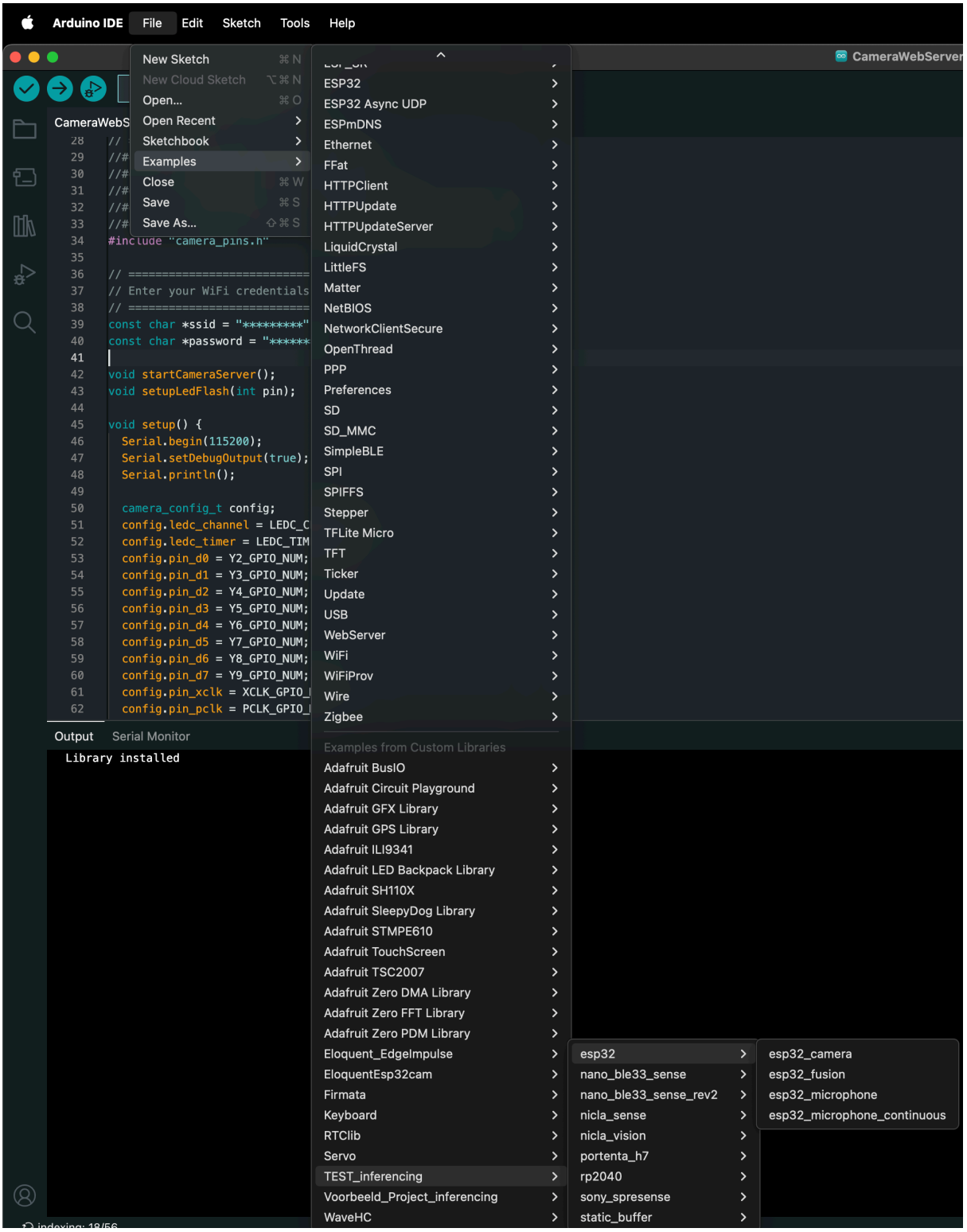
Druk op “Build”, er zal nu een .ZIP gedownload worden die je kan importeren in de Arduino IDE



Doe wat Edge Impulse zegt en druk op Add .ZIP Library in het menu



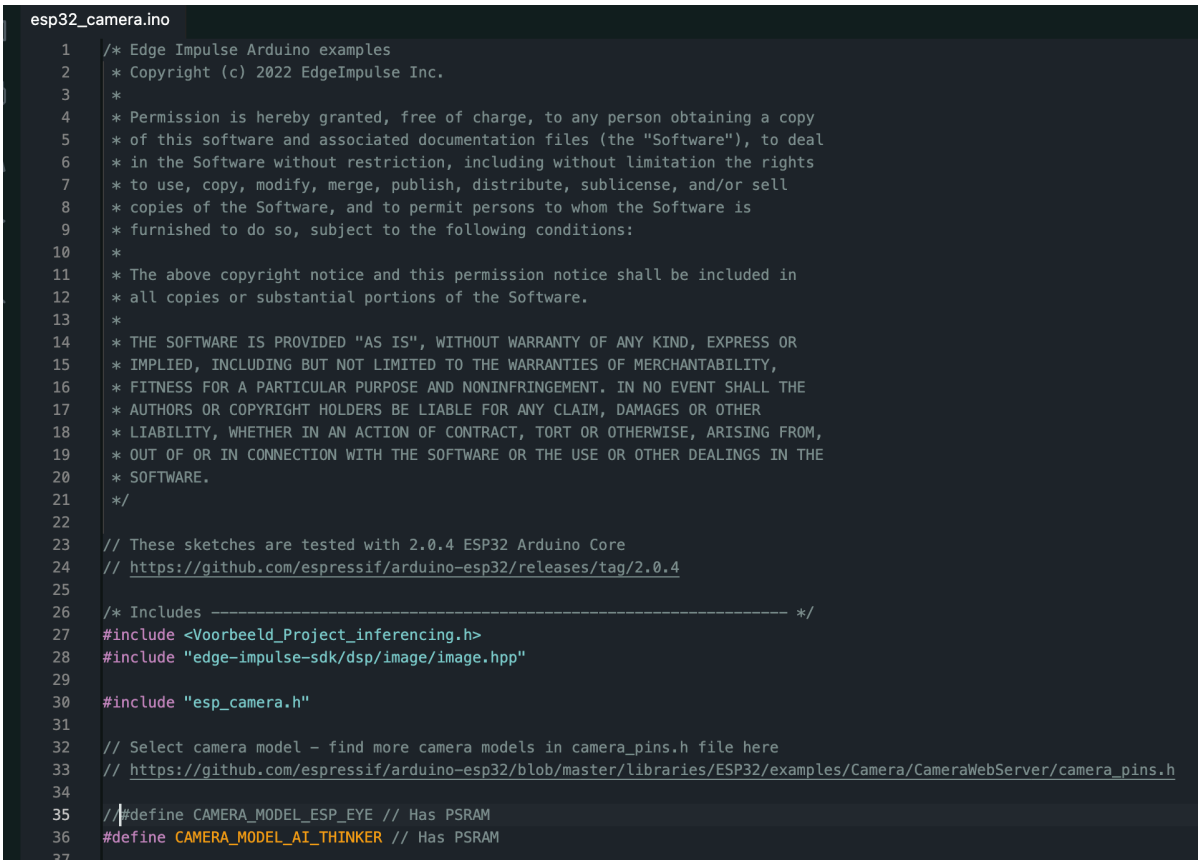
Na het importeren van de Library kan je een Example inladen onder:
File > Examples > Naam van je project (Helemaal onderaan) > esp32 > esp32_camera



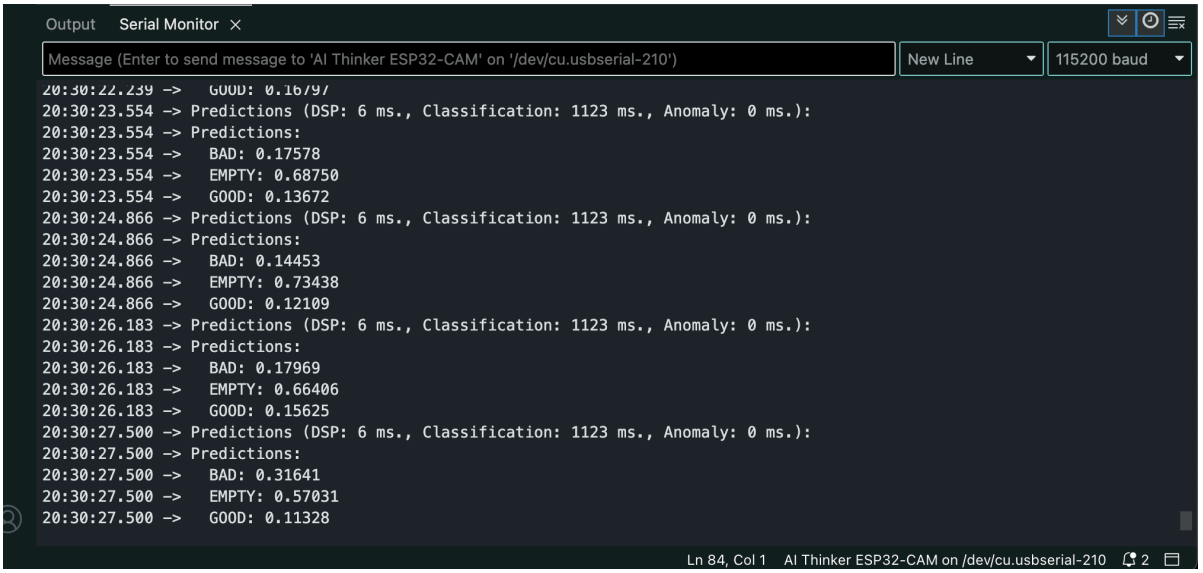
De nieuwe Sketch zal nu openen.

LET OP: op line 36 moeten we nu de juiste ESP selecteren. Dit doe je door

“ `//` ” voor `#define CAMERA_MODEL_ESP_EYE` te zetten en deze slashes weg te halen voor `#define CAMERA_MODEL_AI_THINKER`



That’s it! Nu kunnen we de Sketch flashen naar de ESP zoals we gedaan hebben met de CameraWebServer. Het model zal nu de voorspellingen weergeven in de Serial Monitor.



Deze output wordt geprint op line 190 binnen de `void loop()` van de onaangepaste file (kan mogelijk veranderen).

Binnen de `void loop()` kan je nu aan de slag met het "`result`" om eventueel acties te ondernemen aan de hand van de output van het model.

Gefeliciteerd! Je hebt nu een werkend model op je ESP32-CAM gemaakt met Edge Impulse.