**Opdracht 1: Edge AI met Edge Impulse**

Cijferherkenning

Afbeelding met Graphics, grafische vormgeving, Elektrisch blauw, ontwerp

Automatisch gegenereerde beschrijving

Student: Deklerck Andres

Datum: 18/12/2024

Hogeschool VIVES

Vak: AI Edge (Computing)

Academiejaar 2024-2025

Docent: dr. ing. Jonas Lannoo

Inhoud

[1. Introductie 3](#_Toc186117889)

[2. Aanpak 4](#_Toc186117890)

[3. Gebruikte hardware 5](#_Toc186117891)

[4. Data verzamelen 6](#_Toc186117892)

[5. Preprocessing 6](#_Toc186117893)

[6. Model ontwikkeling 7](#_Toc186117894)

[7. Model testen 8](#_Toc186117895)

[8. Deployment op jouw edge system 9](#_Toc186117896)

[9. Evaluatie, conclusie, future work 10](#_Toc186117897)

# Introductie

Het probleem waarmee ik met dit project wil oplossen, is niet leesbare documenten aan de hand van AI terug leesbaar te maken. De AI kan bepaalde woorden vinden uit de niet leesbare geschreven teksten. Deze kunnen volledig kloppen, of bepaalde letters/cijfers die niet kloppen. Maar het zou alleszins een hele grote hulp zijn hiervoor. Ik wil ervoor zorgen dat het via mijn project mogelijk is om alle letters van het alfabet en de cijfers 0-9 te herkennen.

Het verwachte resultaat is volgens mij een accuraatheid tussen de 70 en 80%. Deze oplossing is interessant voor mensen die graag induiken in oude geschreven documenten. Meestal zijn deze documenten bijna of helemaal niet meer leesbaar, en kan dit project een grote hulp zijn.

Mijn motivatie om dit project te doen stemt uit verschillende kanten, uit de ene kant omdat er veel mogelijke uitbreidingen is voor dit soort project, denk maar aan bijvoorbeeld wachtwoordherkenning. Ook komt het door dat ik vrij geïnteresseerd ben in oude documenten van bijvoorbeeld stambomen. Meestal zijn deze documenten, denk maar aan een geschreven stamboom, geboorteakte, enzovoort niet goed onderhouden, en kan een AI model als deze een grote hulp bieden.

Er zijn zeker limitaties als het project wordt gedaan via Edge Impulse, namelijk hoelang je een model maar mogelijks kan trainen. Een andere limitatie is bijvoorbeeld hoeveel RAM usage het model gebruikt mag niet te hoog zijn, of het zal niet lukken om het te deployen op de hardware. De hardware dat wordt gebruikt in de project is de ESP-EYE 32 ( zie [3. Gebruikte hardware](#_Gebruikte_hardware) ).

# Aanpak

Link naar mijn Edge Impulse project : <https://studio.edgeimpulse.com/public/575476/live>

In de eerste instantie ben ik opzoek gegaan naar een dataset. Deze heb ik gevonden op [Kaggle](https://www.kaggle.com/datasets/dhruvildave/english-handwritten-characters-dataset?resource=download). Dit zijn alle cijfers van 0-9 en alle kleine en hoofdletters van het alfabet. Het zijn een 40-tal afbeelding per klasse.

Hierna heb ik deze data in Edge Impulse geüpload en gelabeld. De split van de dataset voor test en validatie data is 80% voor test en 20% voor validatie.

Na wat problemen ( zie [4. Data verzamelen](#_Data_verzamelen) ) met de dataset, heb ik uiteindelijk besloten om enkel de cijfers van deze Kaggle dataset te gebruiken.

Nu de data volledig in orde stond, ben ik begonnen met de impulse. Hier bevindt zich een Image data block met breedte en hoogte beide 96. De resize mode staat op ‘Fit shortest axis’. Hierna volgt een Transfer Learning block met alle afbeeldingen. In totaal zijn er 10 features ( cijfers van 0-9 ). Hier volgt uiteindelijk natuurlijk de Output features block. Bij de parameters van de ruwe data heb ik ook de ‘Color depth’ ingesteld op Grayscale. Deze afbeeldingen waren al zwarte cijfers op witte achtergrond.

Na het succesvol trainen, komt de Model testing. Eens kijken hoe het model het doet op de test data en nieuwe data. Hierna volgt de deployment van het model op de ESP-EYE 32.

# Gebruikte hardware

Om de data te verzamelen had ik ook eens nagedacht om alle foto’s te nemen met de ingebouwde camera op de ESP-EYE, maar door limitaties en de kracht van de camera heb ik besloten om dit niet te doen. De inference zal ook op die ESP-EYE worden uitgevoerd. Zie hieronder een afbeelding van deze microcontroller.



Omdat ik problemen had met Python op mijn laptop en de verbinding tussen mijn laptop en de ESP, heb ik van de docent een mini Windows PC gekregen om het project hierop verder te zetten.

# Data verzamelen

Eerst heb ik een dataset genomen van Kaggle. Deze dataset bestond uit afbeelding van de cijfer 0-9 en alle letters van het alfabet in klein- en hoofdletter. Er waren rond een 40tal afbeeldingen per cijfer en kleine- en grote letter. Deze data was nog niet gelabeld, dus moest dit nog gebeuren in Edge Impulse zelf.

Ik heb deze dataset dan eens getraind maar de accuraatheid was niet zo goed, rond 60%. De RAM usage was ook veel te hoog om het uiteindelijk te laten runnen op de hardware. Hierna heb ik de dataset verminderd door alle kleine letters uit de dataset te verwijderen zodat enkel de cijfers en de hoofdletters overblijven. Als ik deze trainde met het MobileNetV1 96x96 0.25 architectuur, was de accuraatheid hoger, namelijk 73%. Hier was de RAM usage nog steeds te hoog om het te laten runnen op de hardware.

Ik heb besloten om enkel de cijfers te gebruiken, omdat de RAM-gebruik te hoog was om zowel cijfers als letters te verwerken.

# Preprocessing

In dit project wordt het ‘image’-blok gebruikt. Hier kun je de ‘Color depth’ instellen op RGB of Grayscale. Voor dit project is het niet nodig om met kleuren te werken, dus heb ik het op Grayscale gezet. Dit bespaart ook RAM-gebruik.

Door de ‘Color depth’ op Grayscale in te stellen, bedraagt de verwerkingstijd (Processing time) 15 milliseconden en komt het ‘Peak RAM Usage’ uit op 4 kilobytes.

# Model ontwikkeling

Voor dit project heb ik het MobileNetV1 96x96 0.25-model gebruikt. Dit model bevat geen finale dense layer en maakt gebruik van een dropout van 0.1. MobileNetV1 is een voorgetraind multi-layer convolutioneel netwerk, ontworpen voor het efficiënt classificeren van afbeeldingen.

Het aantal epochs was aanvankelijk ingesteld op 100, maar ik ontdekte al snel dat ongeveer 70 epochs voldoende waren, omdat de prestaties rond dat punt begonnen te stagneren. Ik heb de learning rate meerdere keren aangepast, maar ben uiteindelijk teruggevallen op de standaardwaarde van 0.0005. Daarnaast heb ik data augmentation ingeschakeld. Dit houdt in dat de afbeeldingen worden ingezoomd, gedraaid, gespiegeld, en op andere manieren bewerkt, wat zorgt voor meer variatie in de dataset.

Hieronder ziet u de confusion matrix die werd gegenereerd na de transfer learning-blok. De accuraatheid van de niet-geoptimaliseerde (float32) modelversie bedraagt 83,0%, terwijl de accuraatheid van de gekwantiseerde (int8) modelversie 76,1% is. Dit resultaat is vrijwel goed.

Dit kan men afleiden uit de resultaten en de confusion matrix :

* De klassen 0, 4 en 8 kan het model perfect classificeren. Deze hebben namelijk perfecte scores ( 100% ).
* De klassen 1, 2, 3, 4, 5, 6, 7 en 9 zitten tussen de 60% en 70%. Deze fouten kunnen wijzen op overlappingen tussen deze cijfers in de dataset.
* Klasse 1 is voor 66,7% correct geclassificeerd. De resterende 33,3% worden verdeeld tussen klasse 4 ( 22,2% ) en klasse 7 ( 11,1% ).
* Klasse 2 is maar voor 57,1% correct geclassificeerd. De resterende 42,9% worden verdeeld tussen klasse 7 ( 28,6% ) en klasse 4 ( 14,3% ).
* Klasse 9 is voor 62,5% goed geclassificeerd. De resterende 37,5% worden verdeeld tussen klasse 1 ( 25% ) en klasse 7 ( 12,5% ). Deze fouten verwijzen hoogstwaarschijnlijk naar overlappende kenmerken tussen deze klassen in de data.

Afbeelding met Post-it-briefje, plein, Rechthoek, Kleurrijkheid

Automatisch gegenereerde beschrijving

# Model testen

Het model gedraagt zich zoals verwacht, maar de nauwkeurigheid is lager dan die van de ‘transfer learning’- of ‘retrain model’-blokken. Bij ‘model testing’ is de nauwkeurigheid van modelversie Unoptimized (float32) 61,82%, terwijl de nauwkeurigheid van Quantized (int8) slechts 49,09% bedraagt.

Hieronder ziet u de confusion matrix. Hieruit kunt u afleiden dat het model over het algemeen slechter presteert op elke klasse (elk cijfer). Het slechtst presteert het model op klasse 2 in de testdata.

Wat echter opvalt, is dat het model elk cijfer 1 correct heeft geclassificeerd in de testdata. Volgens mij komt dit door de geringe verschillen binnen de gegevens van klasse 1 in de testdata.

Daarnaast presteert klasse 3 aanzienlijk slechter dan bij Transfer Learning. Het model heeft namelijk slechts twee cijfers correct geclassificeerd als klasse 3.

Afbeelding met tekst, schermopname, plein, nummer

Automatisch gegenereerde beschrijving

# Deployment op jouw edge system

Om het model te deployen op de ESP-EYE, heb ik in Edge Impulse moeten navigeren naar de sectie ‘Deployment’. Bij de deployment-opties heb ik gekozen voor ‘Espressif ESP-EYE (ESP32)’, omdat dit het gebruikte apparaat is. Na het drukken op de knop ‘Build’ wordt er een map gedownload met daarin een bestand waarop je simpelweg moet dubbelklikken. Dit zal het model flashen op het aangesloten apparaat.

Na het succesvol flashen, heb ik in de Command Prompt het volgende commando uitgevoerd:

* edge-impulse-run-impulse –debug

Dit commando zorgt ervoor dat het geflashte model wordt uitgevoerd. De **‘--debug’**-flag maakt het mogelijk om het proces te volgen via een webpagina in plaats van in de Command Prompt.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Hieronder ziet u de webpagina en een voorbeeld. Onder de live webcam ziet u het getal dat het model voorspelt, samen met de nauwkeurigheid tussen haakjes.

Afbeelding met tekst, schermopname, nummer

Automatisch gegenereerde beschrijving

Een volledige opgenomen video is beschikbaar op de volgende repository op GitHub :

<https://github.com/andresdeklerck2023/Edge-Impulse-Opdracht1/tree/main>

# Evaluatie, conclusie, future work

Het project is in het geheel goed gelukt met behoorlijk goede resultaten. Het model draait succesvol op de ESP-EYE en kan de cijfers effectief herkennen.

Ik ondervond een probleem toen ik thuis verder wilde werken aan het project. Ik startte de Windows PC op zoals normaal, maar hij ging niet aan. De aan-knop flikkerde steeds oranje. Dit heb ik opgezocht en de conclusie was dat er een hardwareprobleem was. Hierna heb ik de PC opengemaakt, de knoopcelbatterij uit het moederbord gehaald en na ongeveer vijf minuten teruggeplaatst. Na deze handelingen wilde de computer wel opstarten, maar moest ik een reset doen van de PC.

Om betere resultaten te behalen, kan men meer data gebruiken in plaats van slechts 40 afbeeldingen per cijfer. Daarnaast zou het nuttig zijn om meer variatie in de gegevens aan te brengen.

Voor een uitbreiding van dit project kan men de overige letters van het alfabet toevoegen. Een andere uitbreiding is bijvoorbeeld wachtwoordherkenning. Hierbij schrijft men een combinatie van cijfers en letters op een wit blad, en filmt of maakt een foto van deze code met de camera. Deze code wordt dan vergeleken met het juiste wachtwoord.