

Age Guesser

[Viktor Bouwer Knudsen og Adrian Nordbø], [31.10.2025]

1: BESKRIV PROBLEMET

OMFANG / SCOPE

Våres prosjekt «Age Guesser» har som mål å kunne klare å gjette alderen til en person basert på et bilde av dens ansikt. Vi skal da utvikle en maskinlæringsmodell som skal være tilgjengelig gjennom en enkel nettside, der brukeren laster opp et bilde så skal modellen gjette estimert alder som resultat. Løsningen i vårt prosjekt kan gi en «business impact» ved at den gir brukeren et estimat på alder basert på bildeinput, den kan vise hvordan et nevralt nettverk kan brukes til bildebasert regresjon og ikke minst at vi lærer hvordan man bygger et komplett ML-produkt fra innsamlet data til en nettside.

Dette er en oppgave som ikke lar seg løses effektivt med vanlige programmeringsmetoder, fordi det ikke finnes noen faste regler for hvordan et ansikt skal se ut basert på en spesifikk alder. Ved hjelp av maskinlæring så kan den lære å se mønstre og visuelle trykk som f.eks. rynker, tekstur, form osv. fra et sett med data. Det trengs ikke så mye ressurser for å gjennomføre prosjektet, vi bruker parprogrammeringsmetoden der hvor vi programmerer på 1 datamaskin (alt pushes fortsatt til Git-en vår). Vi bruker Visual Studio Code for å programmere i Python, og et datasett på cirka 10.000 bilder til trening.

METRIKKER

For vårt prosjekt så måles ytelsen hovedsakelig ved hjelp av MAE (Mean Absolute Error) og MSE (Mean Squared Error). Vi bruker disse fordi vår oppgave går ut på å forutsi personens alder, ved hjelp av MAE får vi oversikt over gjennomsnittlig avvik i antall år mellom modellens prediksjon og den faktiske alderen, mens MSE «straffer» større feil, noe som gir et mer nøyaktig bilde av modellens presisjon.

Vi prøvde først ut om det var nok å kun bruke MSE, men fant fort ut at MAE er kritisk for at vi skal få til en god løsning og en god maskinlæringsmodell som løser problemstillingen vår. Vi leste oss opp på GeeksForGeeks

(<https://www.geeksforgeeks.org/machine-learning/regression-metrics/>), her var det godt forklart hvilke metrikker som passet til vårt prosjekt

For selve webapplikasjonen så kan vi måle ytelse i form av responstid, stabilitet og brukertilfredshet. Målet vi har satt oss er at modellen skal i snitt bomme med mindre enn fem år, altså en $MAE \leq 5$. Vi har også et mål om at nettsiden skal levere resultatet på under to sekunder. Disse metrikene støtter prosjektets formål, altså å lage en enkel, engasjerende og lettforståelig maskinlæringsdemonstrasjon der brukeren får et umiddelbart svar.

2: DATA

Vi gikk gjennom og vurderte litt forskjellige datasett som fantes på nett og kom frem til at datasettet til «UTKFace Dataset» var det som passet best til vår oppgave. Vi fant datasettet på Kaggle hvor det sendte oss videre til deres nettside hvor vi enkelt kunne laste ned hele datasettet.

Dette datasettet inneholder 10.000+ bilder av forskjellige personers ansikter, der det er stor variasjon i kjønn, etnisitet og ikke minst alder. Det finnes bilde av personer fra 0 til 116 år. Bildene er lablet med tall som beskriver alder fra 0-116, kjønn der mann = 0 og dame = 1, og etnisitet fra 0-4. Noe som gir hvert bilde filnavnsformat slik: [alder]_[kjønn]_[rase].jpg. Input er RGB-bilder som er fra 6

Vi vurderte om vi skulle bruke et datasett som inneholdt 100.000+ bilder, men konkluderte med at det vil være nok variasjon i alder for at modellen kunne se og lære meningsfulle mønstre uten at treningstiden ble altfor stor. Vi kom over noen utfordringer underveis som vi mistenker er pga datasettet; modellen lærte veldig bra, og under testing gjettet den med mindre enn 5 år ifra riktig resultat, men da vi la inn bilder av oss selv ble det vanskelig for modellen å gjenkjenne mønstre og kunne bomme med opp til 30 år. Vi tror dette er fordi det ikke var nok variasjon med personer med skjegg og briller, noe som kunne gjøre at den gjettet mye høyere eller lavere alder enn det egentlig var.

3: MODELLERING

I dette prosjektet som går ut på å lage en modell som estimerer alder basert på opplastede bilder, valgte vi å bruke Torchvision modellen ResNet-18. Denne modellen er lett tilgjengelig i PyTorch og fungerte bra til vårt prosjekt. Vi gjorde litt research på nett og ved hjelp av Geeks For Geek

(<https://www.geeksforgeeks.org/machine-learning/best-image-processing-tools-used-in-machine-learning/>) fant vi ut at denne modellen passer perfekt til å gjenkjenne komplekse mønstre i bilder. Dette er en standardmodell innen bildeklassifisering og regresjon, noe som ga oss et veldig bra utgangspunkt for å løse problemet vårt.

Vi gjorde en liten endring slik at modellen returnerer et enkelt tall som skal være den predikerte alderen, istedenfor at modellen skulle klassifisere bildene i kategorier. Under trening brukte vi som nevnt tidligere, Mean Squared Error, dette er ofte brukt i regresjons oppgaver så vi valgte også å bruke dette. Etter trening brukte vi Mean Absolute Error for å evaluere modellen ved å få et gjennomsnittlig avvik i antall år.

Etter trening fant vi fort ut at det fantes problemer og feilpredikasjoner, så vi undersøkte litt ekstra på de bildene hvor modellen bommet med mer enn 10 år. Her observerte vi at feilene ofte oppstår på grunn av at personen på bildet har skjegg, briller, dårlige lysforhold eller at personen ikke ser rett mot kamera. Vi prøvde å legge til flere bilder som hadde større variasjon i disse problemene, for å gjøre modellen mer "robust" mot bilder tatt med mobil/pc i nyere tid. Vi trente modellen videre med det oppdaterte datasettet og oppdaget at resultatet ble bedre. Ved testing av personlige bilder traff modellen innenfor 5 år hver gang. Vi fikk en MAE på litt over 3 ved trening.

4: DEPLOYMENT

Vi har tenkt at modellen vår skal fungere som en enkel webapplikasjon, her kan brukeren ta bilde av seg selv ved hjelp av kamera på pcen, eller laste opp et bilde av seg selv og få en alderspredikasjon i retur. For å gjøre dette har vi brukt Gradio, det er raskt og enkelt å fikse et interaktivt grensesnitt direkte fra modulen vår. Når brukeren har lastet opp et bilde, vil applikasjonen automatisk kjøre den samme preprocessering og prediksjonsstegene som den brukte i treningsfasen.

Prediksjonene fra modellen brukes for å estimere en alder på bildet som brukeren laster opp på vår nettside. Dette bildet vil først bli konvertert til riktig format og skalert til riktig piksler før den blir sendt gjennom modellen vår før den så returnerer en aldersverdi som et tall.

Vi tenker at vi setter opp en enkel logging for å overvåke at systemet fungerer som forventet. Dette gjør at vi kan ha oversikt over drift og vedlikehold, ved å se på registrert informasjon som f.eks svartid (latency), tidspunkt og hvilken versjon av modellen som brukes.

5: REFERANSER

Kaggle. (n.d.). *UTKFace dataset (new)*. Kaggle.

<https://www.kaggle.com/datasets/jangedoo/utkface-new>

GeeksforGeeks. (n.d.). *Regression metrics in machine learning*. GeeksforGeeks.

<https://www.geeksforgeeks.org/machine-learning/regression-metrics/>

GeeksforGeeks. (n.d.). *Best image processing tools used in machine learning*.

GeeksforGeeks.

<https://www.geeksforgeeks.org/machine-learning/best-image-processing-tools-used-in-machine-learning/>