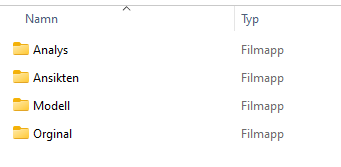
# Egna projektet – Bildanalys

Av: Fredrik Carlsson

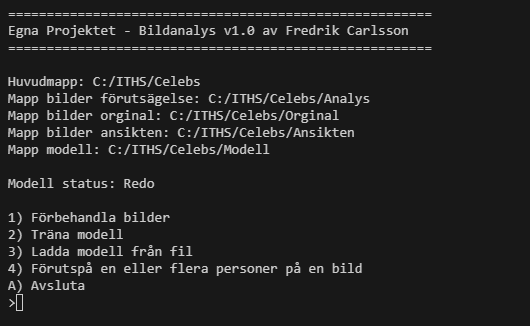
Bildanalys förbereder bilder på kändisars ansikten, tränar en Convolutional Neural Network (CNN) modell och kan därefter analysera en okänd bild med ett eller flera ansikten och förutsäga namn på ansiktena i bilden. Stöd för att spara och ladda en tränad modell.

Det finns en zip fil som innehåller originalbilder, förbereda bilder, sparad modell och okända bilder att analysera. Dessa ligger på mitt ITHS OneDrive konto och kan nås med denna delningslänk: [Bildanalys](https://ithogskolan-my.sharepoint.com/:f:/g/personal/fredrik_carlsson_iths_se/Euy8q33H5lxPrFkOeo27iD4BTtxhpiwScYv0yQeTDp_14g?e=yzis4z)

Packa upp mapparna från zip filen och ange sökväg i Python filen ”start.py”



Kör ”start.py” och använd meny för att komma åt funktionerna i applikationen



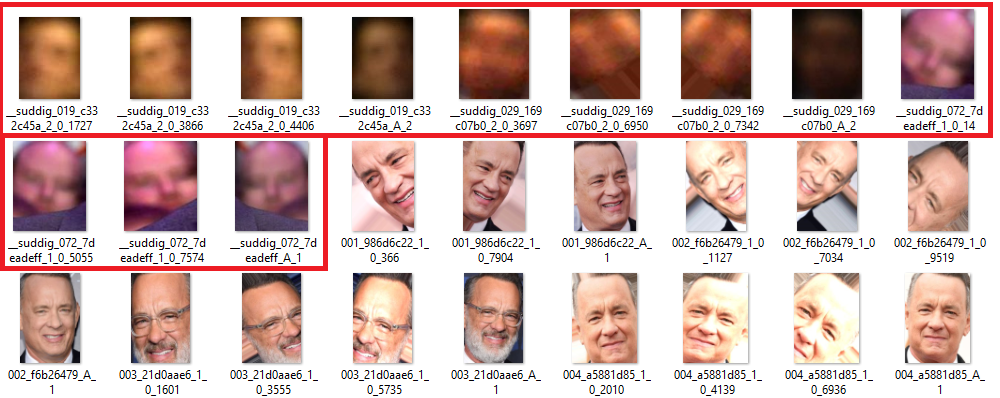
Projektet består av nedan delar:

1. Förbehandling av en eller flera (märkta) bildmappar.
2. Träning av bilder som förberetts
3. Förutsägelse av en eller flera personer i en okänd bild
4. Spara och ladda modell

## Förbehandling

En procedur itererar en mapp med undermappar där varje undermapp anger kategori namn dvs namn på person. För varje bild används Multi-Task Cascaded Convolutional Neural Networks (MTCNN) för att känna av ansikten. Varje ansikte beskärs ut enligt en box som MTCNN hittar och sparas ner som en egen fil. För att förbättra igenkänning, framförallt på kvinnliga modeller med långt hår utökas boxen med 20% om det är möjligt, dvs bildens yttre gräns inte överskrids. Som standard hämtas max 5 ansikten ut från en bild, ansiktenas sorteras efter confidence (nivå på förtroende/tillförlitlighet) så att de bästa ansikten hämtas. Dessutom exkluderas bilder som underskrider minimum upplösning 2048 pixlar.

För att underlätta hanteringen räknas ansiktets suddighet ut genom att räkna ut varians av laplacian (skarpa kanter). Vanligen brukar värde under 100 indikera suddighet men en del bra bilder föll då bort så jag kombinerade < 50 samt confidence < 99% för att tagga bilder som suddiga. Detta genom att lägga till \_\_suddig som suffix till filnamn.



Exempel på suddiga bilder, de syns först i Utforskaren och går enkelt att söka upp. Dessa bilder bör tas bort. Dessutom bör övriga bilder inspekteras, ev bilder på andra personer eller andra objekt måste tas bort för att inte påverka träning negativt.

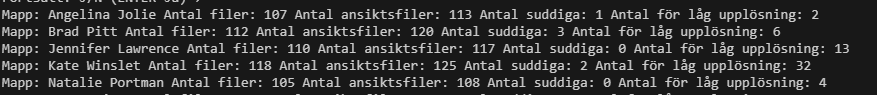
Inspiration till suddighet/laplacian från: https://pyimagesearch.com/2015/09/07/blur-detection-with-opencv/

Augmenterade versioner av bilden skapas också. Följande augmenteringar görs:

* Rotera (upp till 60gr)
* Horisontell förflyttning
* Ändring av ljusstyrka i intervallet 0,5 – 1,5 (minus 50% till plus 50%)
* Zooma i intervallet 0,7 – 1,0

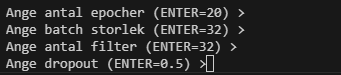
De icke augmenterade bilderna har ett \_A\_ i slutet av filnamnet så att man enkelt kan göra träning enbart på icke augmenterade bilder. Under träning gav de augmenterade bilderna ett sämre resultat av oklar anledning.

När programmet körs visas statistik för varje bildmapp



## Träning av bilder som förberetts

Träning görs med 80% av data från ansiktsmapp som förberetts av förbehandlingen. Vid körning är det möjligt att ange nedan parametrar. Tryck ENTER för att acceptera standardvärden

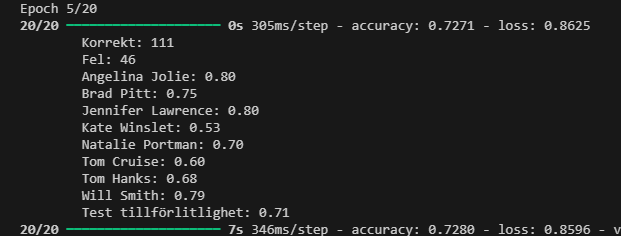


Bilderna laddas in och normaliseras till storlek 224x224, pixlar reduceras till värde mellan 0..1 och kategorier encodas.

CNN modell skapas, som input anges shape/storlek på bilderna, dropout används för att undvika overfitting genom att ta angiven procent neuroner. Softmax avslutar och räknar ut andel andel sannolikhet per kategori.

Inspiration till CNN: <https://learnopencv.com/implementing-cnn-tensorflow-keras/>

Mellan varje epoch visas totalt korrekta och fel av test andelen samt noggrannhet per kategori



Efter träning kan man spara ner modell till fil. Kategorier sparas i en egen textfil med samma namn

## Förutsägelse av en eller flera personer i en okänd bild

Välj en av de jpg filer som finns i mappen ”Analys”. Ett plot fönster visar de ansikten som hittats, om träffsäkerheten är >70 visas namn och träffsäkerhet. Annars visas inget namn

Med den bifogade filen ”DetEgnaProjektet” som uppnådde ca 75% test resultat känner modellen av nedan personer på de bifogade okända bilderna i analys mappen.

Bild ”natalieportman\_bradpitt.jpg”

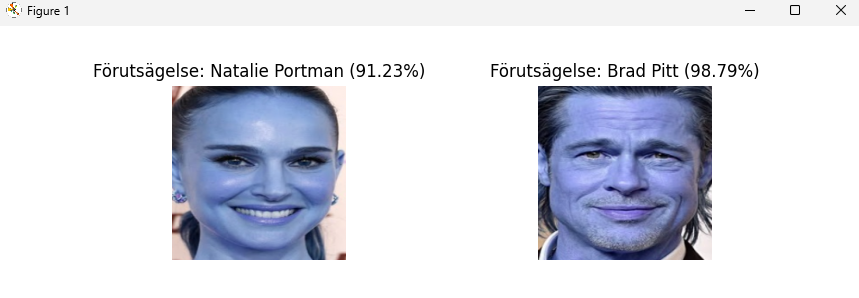
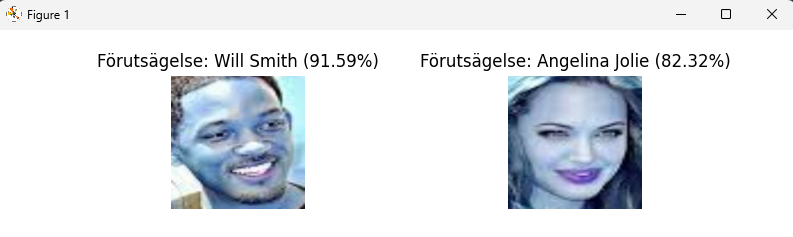


Bild ”willsmith\_angelinajolie.jpg”



## Vald av algoritm

Jag valde att använda CNN algoritmen då den gav goda resultat med relativt små mängder data, den är dessutom relativt snabb på en vanlig PC. För nybörjaren gick det snabbt att hitta bra resultat utan för mycket krångel.

## Använda bibliotek som behöver installeras för att köra applikationen

MTCNN

Tensorflow

Sklearn

CV2

Numpy

Matplotlib

MTCNN installeras med ”pip install mtcnn”