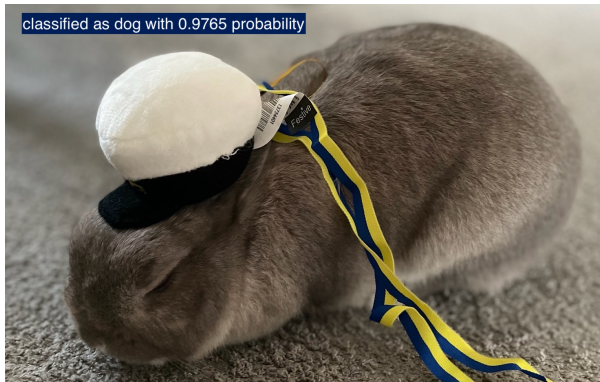


# Laboration deep learning

---

**Syftet** med den här labben är att använda deep learning för computer vision för att lära en algoritm att klassificera olika objekt.



I den här labben kommer du få jobba med bildbehandling, filhantering, bygga upp egna tränings, validerings och testdataset från en mängd av bilder. Vidare får du jobba med begränsade datamängder för att efterlikna verkligheten när dataanskaffning är en oerhört dyr process. Efter databehandlingen och databearbetningen kommer du få pröva olika slags convolutional neural networks tränade från scratch och slutligen använda dig av transfer learning för att återanvända nätverk skapade av toppforskare inom computer vision.

---

## Uppgifter

Dokumentera i markdown i Jupyter notebook eller i separat markdownfil om du använder Pythonskript.

---

### 0. EDA och filhantering (\*)

a) Börja med att ladda ned datasetet [Dogs vs Cats](#) från Kaggle. Extrahera den och lägg den i din labfolder, glöm inte att lägga till den till .gitignore.

b) Läs in 10 bilder slumpmässigt, plotta dem och extrahera deras labels och skriv ut i titeln.

c) Skapa folderstrukturen nedan med Python:

```
.
├── experiment_small_data
│   ├── test
│   ├── train
│   └── val
├── experiment_tiny_data
│   ├── test
│   ├── train
│   └── val
├── lab.ipynb
└── original_data
    ├── test
    └── test
```

```
└─ train
  └─ train
```

Notera att det är okej att ha fler pythonskript, notebooks, fler experiment, eller annat relevant i labbfoldern. Glöm inte att ha gitignore på datafilerna.

d) Nu ska du göra train|val|test split med följande splits:

experiment\_small

- train - 1600 (800 dogs, 800 cats)
- val - 400 (200 dogs, 200 cats)
- test - 500 (250 dogs, 250 cats)

experiment\_tiny (BONUS)

- train - 160 (80 dogs, 80 cats)
- val - 40 (20 dogs, 20 cats)
- test - 50 (25 dogs, 25 cats)

Det är inte så farligt om det inte blir exakt balanserade dataset.

Det är viktigt att du samplar slumpmässigt utan replacement från originaldatasetet under respektive experiment. Spara datan i deras respektive mappar. (\*\*)

e) Läs in dataseten från experiment\_small, experiment\_tiny (BONUS) och plocka ut labelsvektorer, som ska vara one-hot encoded med 0 och 1.

- plotta några bilder med deras respektive labels och kontrollera att det är korrekt.
- skapa lämplig plot för att kontrollera att dataseten är balanserade
- skapa lämplig plot för att kontrollera att dataseten är slumpade, dvs inte ex

```
[0, 0, ... 0, 1, 1, ..., 1]
```

---

## 1. Bildbehandling (\*)

Uppgifterna nedan ska upprepas för respektive experimentdataset om du också väljer att arbeta med tinydatasetet.

a) Skapa en plot för att visualisera bildstorlekarna i träningsdatan. Använd seaborns jointplot.

b) Välj en lämplig bildstorlek att ändra samtliga bilder till. Gör en analys och fundera på om du behöver slänga bilder. Hur kommer du fram till ditt val?

c) Gör resize sådant att samtliga bilder är samma storlek och spara dem i numpy arrays med följande struktur:

```
(samples, row, cols, color_channels)
```

Visualisera därefter ett par styckena bilder.

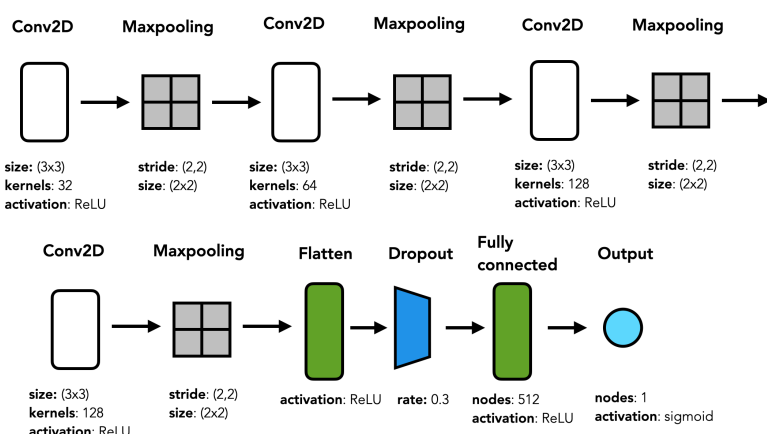
d) Augmentera datan. Varför behövs dataaugmentering och hur beslutar du vilka parametrar du valt för augmenteringen?

## 2. Träna modeller (\*)

För uppgifterna nedan, jobba med dataseten en åt gången:

- small
  - utan augmentering
  - med augmentering
- tiny (BONUS)
  - utan augmentering
  - med augmentering

a) Använd följande nätverk och träna på datan. Gör hyperparametertuning för några parametrar (beskriv hur du gör). Visualisera och analysera loss-kurvor, accuracy-kurvor.



b) Förändra nätverket i a) experimentera och ändra i lite komponenter. Beskriv vad du ändrar och varför. Glöm inte att evaluera på valideringsdatan.

c) Välj en modell, träna på tränings- och valideringsdatan. Gör inferens på testdatan och utvärdera din modell.

d) Tag ett eller flera av nätverken ([VGG16](#), [Resnet](#), [Xception](#), [Inception](#)), läs deras forskningsartikel, sammanfatta kort ca 1/2 - 1 sida för en artikel. Utför därefter transfer learning och evaluera din modell. Beskriv också hur transfer learning fungerar.

## Bedömning

Om du har fått någon kodsnuitt från någon annan, LLM såsom ChatGPT eller hittat i någon sida är det **viktigt** att du källhänvisar. Skriv en kommentar bredvid koden som du har tagit.

Godkänt

- löst uppgift 0-2 på ett korrekt sätt

- koden är kommenterad med relevanta kommentarer
  - har beskrivit och motiverat val av parametrar, modeller mm
  - teorifrågorna är korrekt besvarade
  - filhanteringen, bildbehandling, EDA har gjorts korrekt
  - variabelnamnen är bra valda
  - gjort flera relevanta git commits
- 

## Väl Godkänt

Uppfyllt allt för godkänt samt:

- delarna märkta med BONUS är gjorda
- koden är tydlig och enkel att följa
- koden är välstrukturerad med funktioner och/eller OOP med hög grad av DRY
- teorifrågorna är besvarade med hög nivå
- du motiverar koden väl med datavetenskapligt korrekt språk
- du motiverar väl dina val av parametrar, modeller mm