AI-ML-principer för datadrivna organisationer - Inlämning 1

I denna rapport beskrivs det följande delar av en maskininlärnings process:
1. Val av dataset
2. Syftet och mål med dataset
3. Grundläggande beskrivning av dataset
4. Träning och evaluering av olika modeller
5. Slutsats

Márk Mészáros

1. Val av dataset

För denna uppgift valde jag en dataset från Activeloop. Datasetet heter CODEBRIM (https://app.activeloop.ai/datasets/search?query=luizapzbn%2FCODEBRIM&sortby=stars&sortdir=desc).

Datasetet innehåller bilder av väggar och olika strukturer byggt av betong, och visar båda oskade och skadade ytor.

2. Syftet och mål med dataset

Syftet med datasetet är att använda det för att träna upp en modell som kan classifiera dom olika typer av skador på betongytan, så vi kan effektivisera skadebesiktning och skadekontroll av större byggnader och objekt. Istället fr att manuellt underska en bygnad, potenciella skador kunna identifieras innan manuellt besiktning görs.

Primära målet är att ge en tydlig bevis om att det går att träna upp en modell som kan skilja olika typer av skador på betongytan, med 80%+ punktlighet.

Alternativa målet är att ge en tydlig bevis om att det går att träna upp en modell som kan skilja mellan oskadade och skadade betongytor, med 80%+ punktlighet.

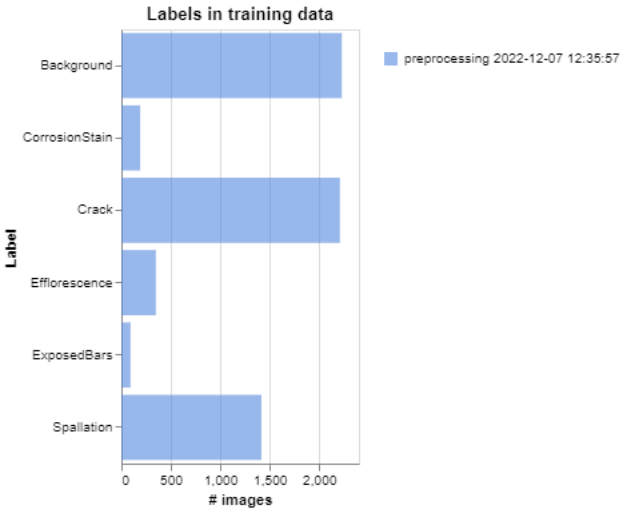
3. Grundläggande beskrivning av dataset

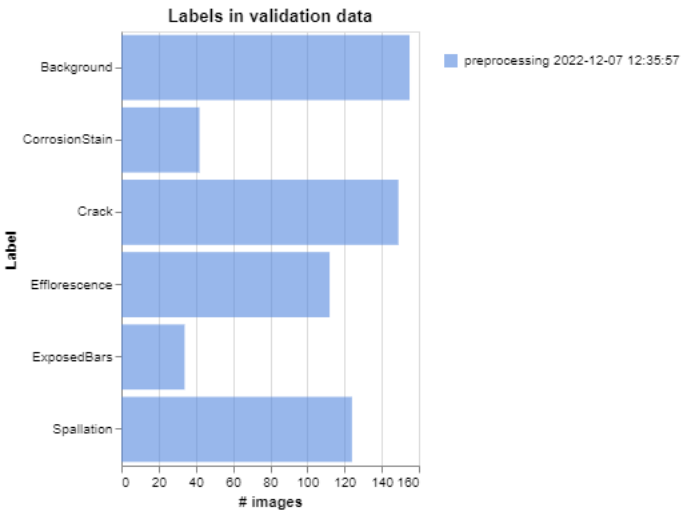
3.1 Kategorier

Träningsdatasetet består av 6481 biler, och valideringsdatasetet består av 616 bilder.

Bilderna har 256 X 256 pixlar och har 3 channels (RGB).

Båda datasets har det samma kategorier:






```
runs.summary["Labels in training data_table"]
```

	Label	# images
1		
2		
3		
4		
5		
6		

1 - 6 of 6 Export as CSV Columns... Reset Table

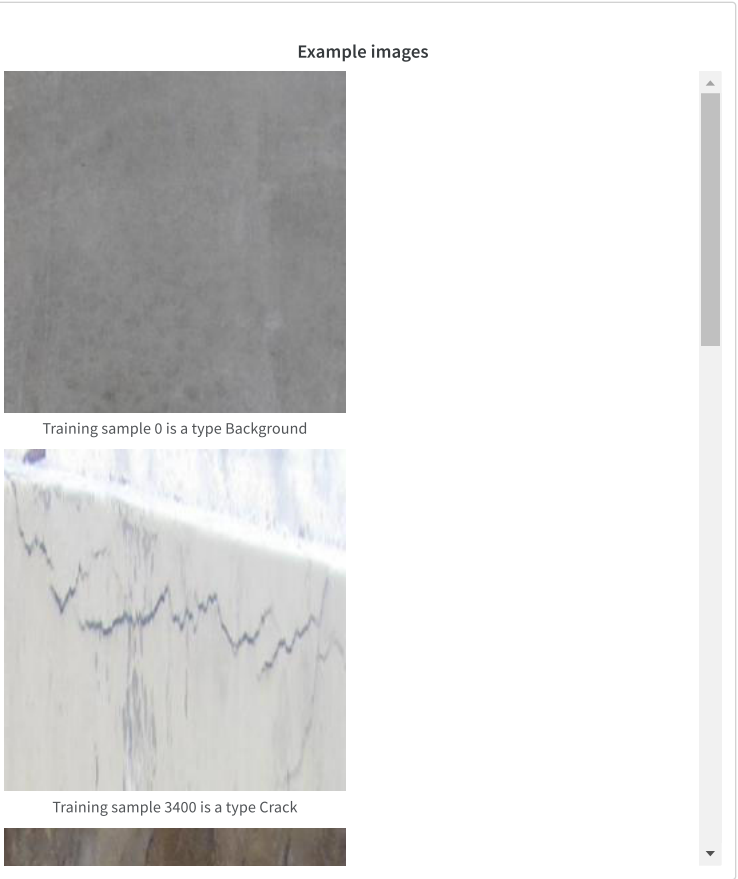
```
runs.summary["Labels in validation data_table"]
```

	Label	# images
1		
2		
3		
4		
5		
6		

1 - 6 of 6 Export as CSV Columns... Reset Table

Dom olika kapegorier med exempel:

'Background', 'Crack', 'Spallation', 'Efflorescence', 'ExposedBars', 'CorrosionStain'



Anmärkning:

Bildarna ovanpå kommer från originella datasetet, men under processen har jag både gjort förändringar på bilderna och på deras

etiketter.

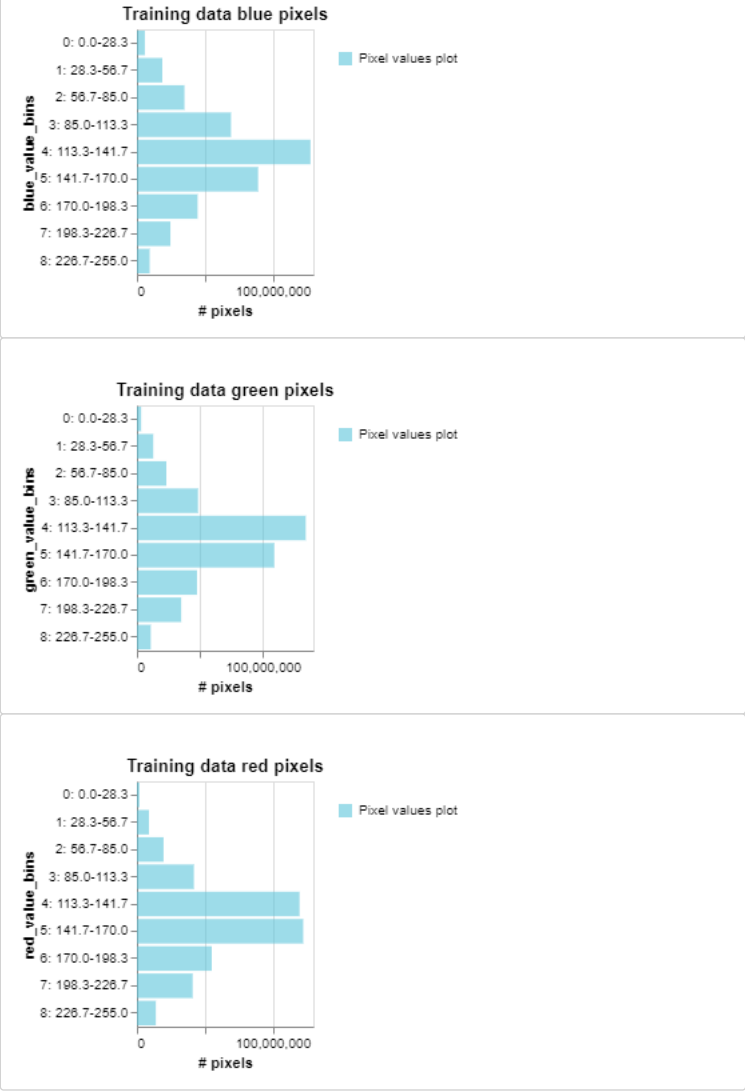
I model träningen har jag använt RGB och gråskaliga bilder. Mer information om detta under punkt 4.1 och 4.2.

Jag har använt dom ursprungliga 6 kategorier för klassificering av bilder, men har förändrat etiketter till "Skadade", "Oskadade" senare.

3.2 Pixlarnas värde och fördelning

Histogram under representerar pixlarnas värde och dess fördelning för RGB bilder.

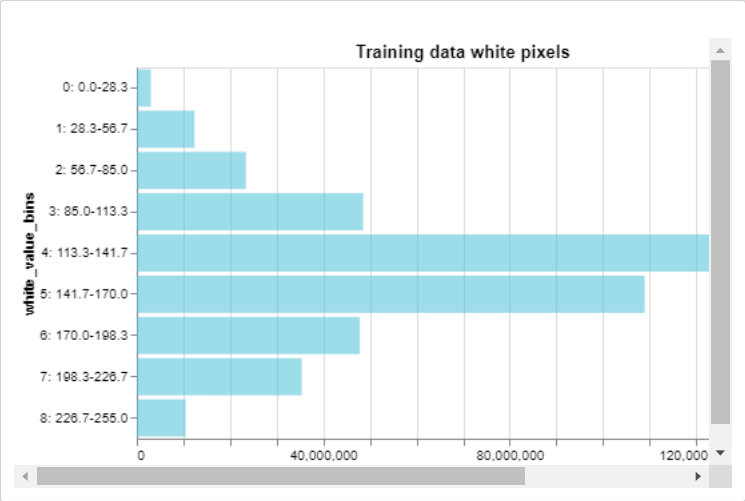
Som vi ser från histogrammen, pixelvärdernas fördelning är inte alls uniform. En stor majoritet av pixlarna har högra värde, mellan 100 och 200, vilket betyder att bilderna är huvsatt ljusa, och väldigt få pixlar är inte alls eller fullt mättad.



Följande histogram representerar pixlarnas värde mellan 0 (svart) och 255 (vit) på samma bilder, men nu har jag transformerat dom till gråskaliga bilder.

Vi ser väldigt liknande fördelning på pixlarnas värde; få pixlar är helt svarta, eller mörka och få pixlarna är vita. Mest av pixlarna har en värde mellan 100 och 200.

Detta är en rimlig påstående eftersom bilderna visar (skador på)



3.3 Potenciella problem med dataset

Efter en snabb genomgång på bilderna i datasetet, jag har ifrågasatt giltigheten av märkningar vissa bilderna har.

Problemet är att vissa skador skulle kunna passa i en annan eller till flera kategorier. Några exemplar:

Detta problem kan i stor utsträckning påverka vårt modells prestanda på en negativ sätt, eftersom klassificering av skador blir inte självklart och olika mnstrar i datan kan överlappa.

Förslag för att lösa problemet är det följande:

- Märka om bilderna med hjälp av specialister som har kunskap i domänen

- Ta bort bilder med tvetydiga märkning
- Klassifiera om bilder som kan överlappa / bli förvirrad med varandra till en gemensamma klass
- Klassifiera om bilder till skadade-oskadade märke

På grund av tids-; och kunskapsbrist inom domänen valde jag att klassifiera om bilder till skadade-oskadade märke, men kommer ändå göra några försök att träna modeller på ursprungliga märker.
En annan potenciella problemkälla är att antal bilder i kategorierna är inte lika. Vissa av kategorierna är överrepresenterad. Kategorierna CorrosionStain, Effloresence och ExposedBars är grovt underrepresenterade i båda tränings och valideringsdataset, och Background, Crack och Spallation är överrepresenterad. Detta kan leda till att dom underrepresenterade kategorier blir underrepresenterad i valideringen, eftersom modellen har inte träffat dom så ofta för att upptäcka ett mnster och kan dessutom (med hög accuracy) gissa att en bild inte är från den kategorin. Detta problem skulle kunna behandlas genom att:

- Sammla in flera bilder från underrepresenterade kategorier, så kategorifördelning blir mer jämt
- Jämna ut validerings datasetets kategorifördelning genom att minska antalet bilder från överrpresenterade kategorier
- När vi räknar accuracy, så skulle vi kunna vikta om resultatet baserat på kategoriernas fördelning

4. Träning av olika modeller

Vår problem är en klassifikations problem: vi har 6 olika klasser som bilderna ska delas in i. Detta redan utesluter några modeller ur paletten.

Våra bilder är märkta (labeled) vilket innebär att vi kan ta njutta av en Supervised learning modell: vi tillhandahålla kategorien för varje bild, och modellen kommer att fundera ut ett mönster, en regel för att klassifiera träningsbilderna och applicera det samma regel senare när vi visar ett nytt bild till modellen.

Jag har provat det följande modeller:

- Random Forest

- Convolutional Neural Network

4.1 Random Forest

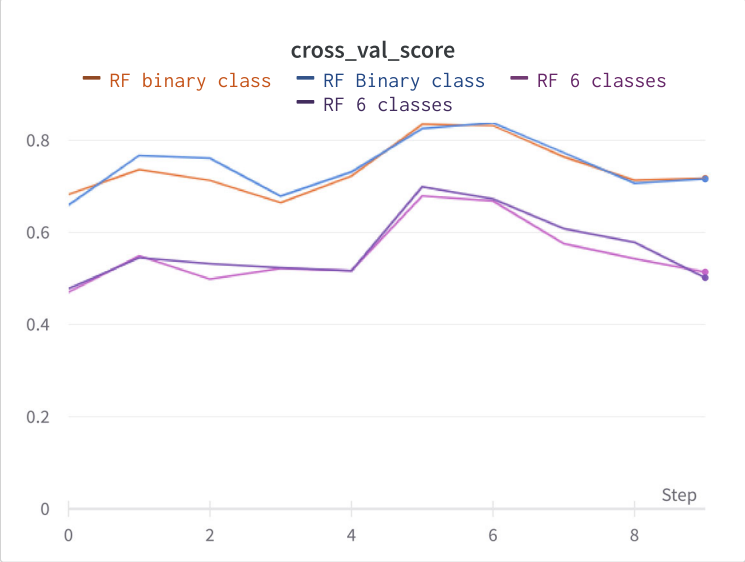
Ranom forest modellen fingerar bara med 2 dimensionella arrays, vart dimension 1 representerar dom olika sämplar och dimension 2 representerar en gråskaliga bildens alla pixelvärde i ett långt rad.

Tack varde detta egenskap, för Random Forest modellen har jag ändrat alla bilder till gråskaliga.
I nedanstående plotten kan vi se RF modellens accuracy som jag har testat med sklearn.cross_val_score. Detta metod (i min fall) delar tränings datasetet till träning och validerings delar (90%-10%), tränar

och evaluerar modellen 10 gånger och mäter modellens accuracy 10 gångar.

Som vi kan se, vi får betydligt högre accuracy när vi använder binary

Random Forest model accuracy



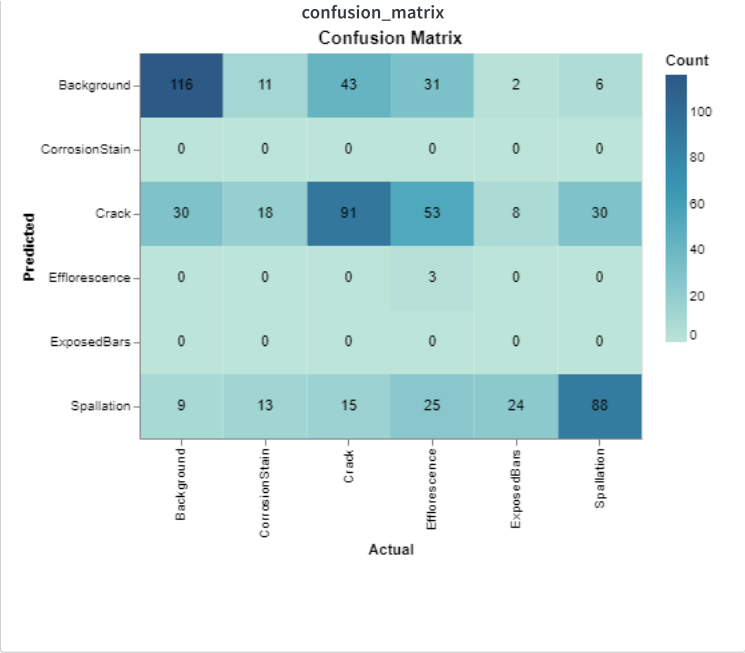
I vårt fall är accuracy inte det bästa metric, eftersom vårt validations dataset ät lång ifrån uniform fördelat, vilket innebär att modellen kan nå hög accuracy genom att gissa att bildarna är i klass Background, Crack eller Spallation, och gissa att ingen av bilderna är klassade som CorrosionStain, Efflorescence, eller ExposedBars.

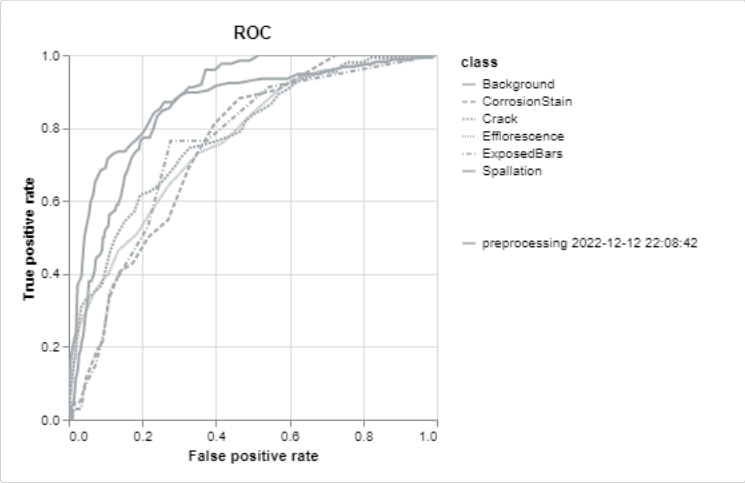
Jag har därför användt Confusion matrix, ROC kurva och Precision-Recal kurva som metric.

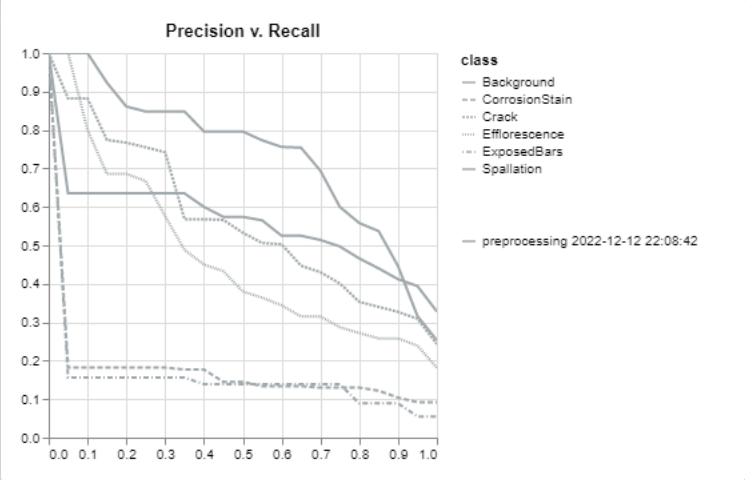
- Confusion matrix: diagonalen visar hur många kategorier modellen har gissat rätt, och dom andra rutor visa hur många bilder har vi kategoriserat felaktigt som en annan kategori.
- ROC kurva:
- Precision-Recal kurva: grovt förenklad vill vi se att kurvan närmar sig till x=1.0, y=1.0 punkten. Ju närmare kurvan löper det desto bättre är vi att klassificera rätt bild till rätt klass. I en ideal fall (när kurvan går genom x=1.0, y=1.0 punkten) så har modellen gissat alltid rätt klass (precision (y) = 1 --> 100% av t.ex. Background gissningar var faktiskt Background) och alla bilder med en viss klass har hitats (recall (x) = 1 --> 100% av bilder med sant klass t.ex. Background blev klassifierad som Background).

4.1.1. Random Forest med 6 kategorier

Vi kan observera att vårt model är bäst på att identifiera Background bilder: Båda ROC och PR kurvan är närmast till ideal kurvan.
Vi kan se att våra förväntningar har uppfyllts och vi ser en tydlig överrepresentation av Background, Crack och Spallation kategorierna, till nackdel för dom andra tre kategorier som blev inte rätt kategoriserad av vårt model.





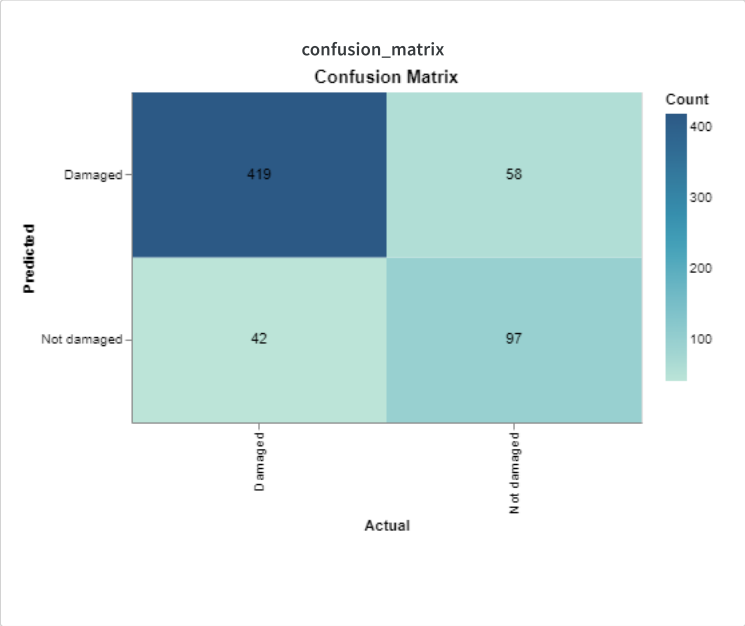


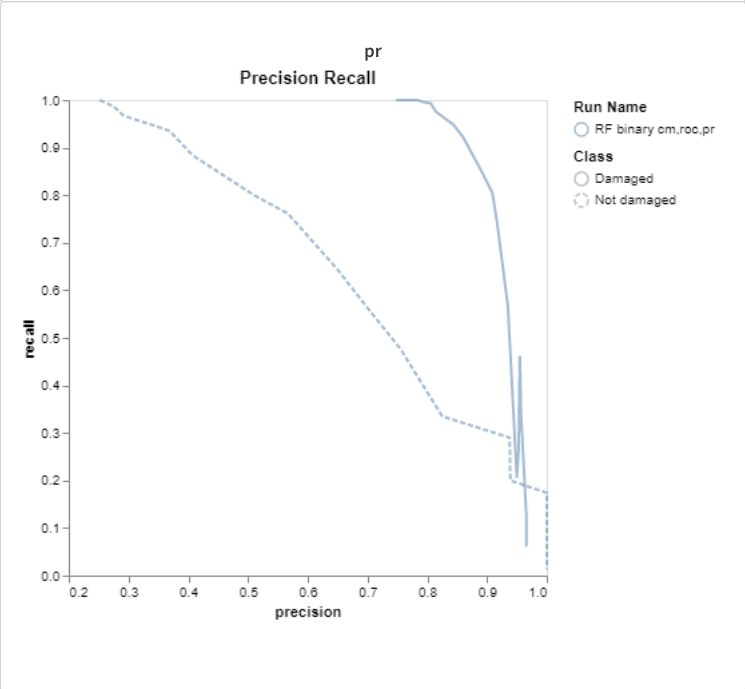
4.1.2. Random Forest med 2 klasser

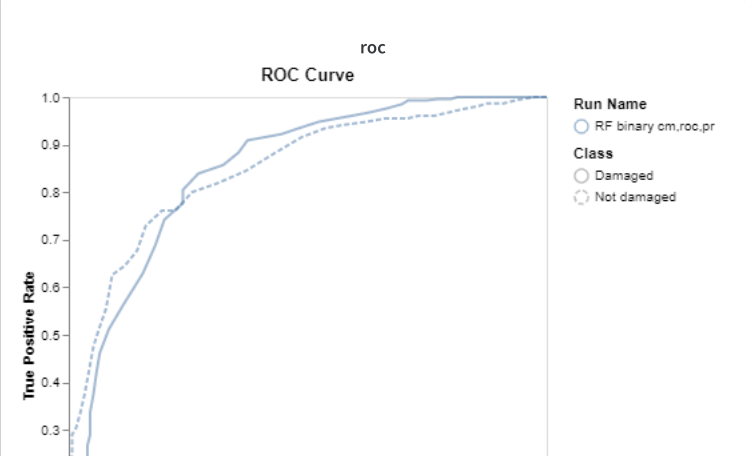
Här kan vi se att modellen är bättre med att identifiera Damaged och Not damaged klasser. För vårt use-case är det viktigt att upptäcka potenciella skador på betongen och att ha några False Positives är

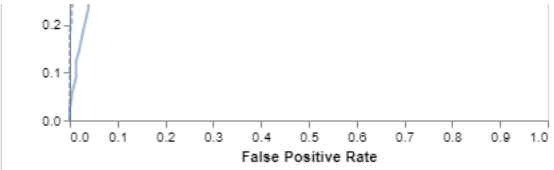
ingen sor fara, eftersom båda inspelningen kan kontrolleras av en människa och ytterligare undersökningar kan göras av alpinister.

Om vi tittar på PR kurvan så ser vi att modellen är bra med att fånga bilder på skadade ytor, och har en relativt låg False Positive Rate.









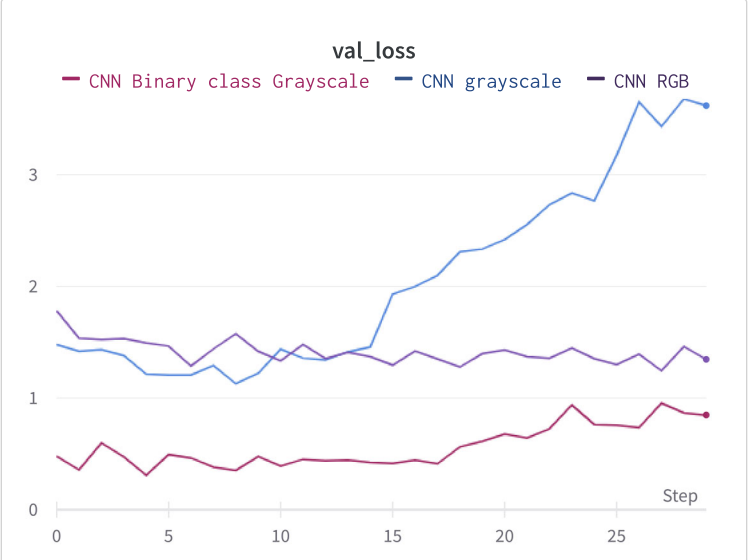
4.2 Convolutional Neural Network

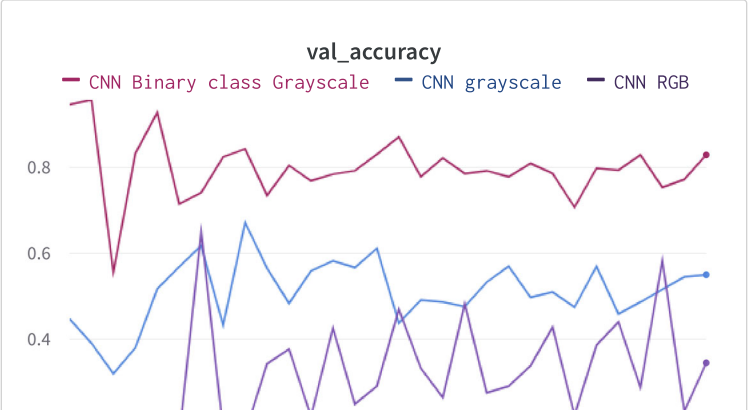
CNN är en av dom vanligaste modeller för att jobba med bilder i en klassificerings-; och detektionssyfte.

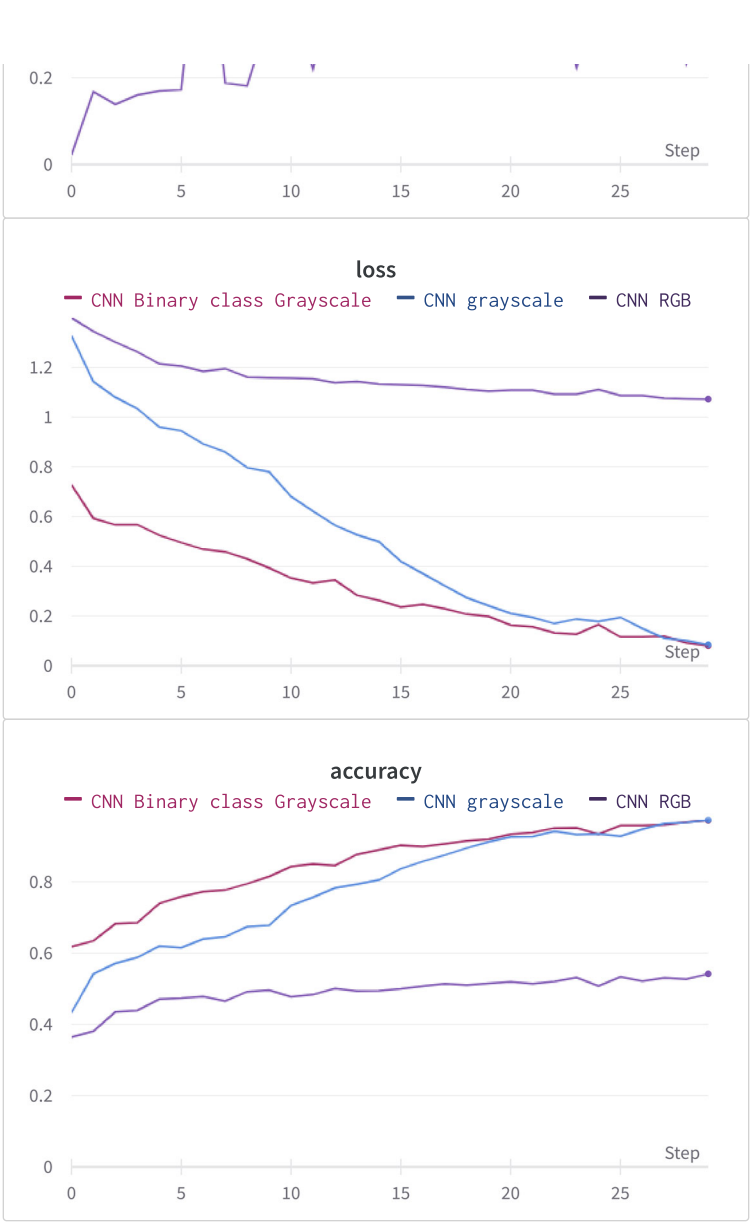
Nedan kan vi se båda in-training och validation accuracy och loss funktionen för modellerna.

Dom 3 grafer representerar 3 olika tränade modeller, med olika träningsdata:

- Lila grafen representerar modellen som var tränad på RGB bilder och 6 klasser.
- Blåa grafen representerar modellen som var tränad på gråskaliga bilder och 6 klasser.
- Röda grafen representerar modellen som var tränad på gråskaliga bilder och 2 klasser.







Vi kan se att RGB bilderna var det svåraste för modellen, och den hade inte särskilt högt accuracy med att bestämma rätt kategori för rätt bild. Accuracyn under träningen har nått en platå och validations accuracy var mycket varierande. Detta skulle möjligen förbättras med flera träningsdata, och data augmentation.

Modellen har hanterat gråskaliga bilder bättre än RGB bilder, vi kan se en tydlig ökning av accuracy och minskning av loss under träningen. Även om validations accuracy är bättre än med RGB bilder, men det är också mycket varierande, och när vi tittar på validation loss grafen, där ser vi en tydlig ökning, vilket betyder att vi har Overfitted vår model.

Modellen som har ska identifiera skadade och oskadade väggar på gråskaliga bilder har fungerat bäst. Validerings accuracy var högst med denna modell, och validation loss var minst. Det ser så ut att modellen kunna ge det bästa resultat inom mindre än 10 träningsvarv / epoch.

5. Slutsatser

Till slut ser det så ut att vi kunde skapa två modeller som är bra på att klassifiera våra bilder, men det verkar se ut att bästa resultatet fick vi med gråskaliga bilder och när vi hade bara två klasser (Damaged - Not damaged).

Ytterligare tester skulle behövas för att säkerställa båda RF och CNN modellens accuracy på en ny test set, men vi har nått vårt ursprungliga mål, och denna lösning med att identifiera binära klasser skulle också passa till vårt use-case.

Frågorna till fortsättningen är vad våra tekniska frutsättningar är, eftersom CNN modellen kan i optimalt sätt hantera RGB bilder vid fortsätt optimalisering, och det går att använda GPU accelererad model träning och prediction vilket kan resultera i snabba lösningar om det skulle krävas, medans RF modellen är mer enkel, kan bara hantera gråskaliga bilder och stöder ingen GPU acceleration.

Optimalt sätt skulle vi behöva mera bilder med bättre märkningsprecision för träning, validering och test datasets, och augmentera träningsbilder så vi skulle kunna bygga en mer robust model.