Taggning av bygglovshandlingar

# Behovet av taggning

Ett av målen med detta projekt har varit att ta fram en eller flera AI-modeller som kan avgöra om en bygglovshandling innehåller något vanligt förekommande misstag, för att snabbt kunna ge denna feedback till användaren.

Inom ramen för detta projekt fokuserar vi på att kunna hantera fyra olika typer av bygglovshandlingar: situationsplaner, nybyggnadskartor, fasadritningar och sektionsritningar. Vi letar efter ett antal olika objekt som ofta blir felaktiga eller saknas i ritningarna: skalstock, kompassriktning, avstånd till tomtgräns, marklinjer och fasader. Varje typ av bygglovshandling ska enligt reglerna innehålla en specifik delmängd av dessa objekt. Objekt kan också se ut på olika sätt. Kompassriktning anges ibland med en kompassros och ibland i textform, till exempel “Fasad mot norr”.

En AI-modell kan tas fram på olika sätt. Det vanligaste sättet är så kallad övervakad inlärning, där en modell under en träningsfas får se ett antal exempel och förväntade svar. Till exempel kan en modell få se en ritning tillsammans med extra information som anger exakt var kompassrosen finns, eller en ritning tillsammans med information som anger att det saknas en skalstock. Modellen lär sig då att koppla ihop ritningen med den extra informationen och kan sedan användas för att avgöra om en ny ritning innehåller något fel.

Ett annat syfte med att ta fram ett antal exempel med korrekt upptaggade ritningar är att dessa sedan kan användas för att utvärdera hur väl modellen presterar. Då kan man jämföra modellens svar med de korrekta svaren och se hur många fel modellen gör.

För träning och utvärdering behövs alltså ett antal exempel på ritningar som är korrekta och ett antal exempel på ritningar som innehåller fel. Dessa exempel måste vara upptaggade, det vill säga att det måste finnas information om vilka objekt som finns i varje ritning och vilka objekt som saknas. Detta är en tidskrävande process som kräver att någon manuellt går igenom varje ritning och markerar de olika objekten. Detta görs bäst av en specialist som är van att handlägga och granska bygglovshandlingar.

Denna upptaggning måste också göras på ett mycket konsekvent och strukturerat sätt, så att modellen korrekt kan lära sig av datan. Därför används ofta ett speciellt anpassat gränssnitt, så att det är lätt att markera objekten i ritningarna och att datan sparas på ett konsistent sätt.

# Val av taggningsverktyg och installation

Vi har använt Label Studio, https://labelstud.io/, som annoteringsverktyg.

Label Studio valdes för att det har öppen källkod, gott stöd för att annotera bilder, samt stöd för flera användare som hjälps åt med en uppgift.

Label Studio laddas enklast ned och körs med docker:

docker run -it -p 8080:8080 -v `pwd`/mydata:/label-studio/data heartexlabs/label-studio:latest

Detta kör senaste versionen, vilket i vårt fall var 1.11.0, men det kan rekommenderas att välja en specifik version via en annan tagg än “latest”, så att inga oväntade uppgraderingar sker.

Eftersom verktyget skulle vara åtkomligt för både Tenfifty och Kungsbackas bygglovsexperter valde vi att köra det på en virtuell maskin i molnet. En statisk IP-adress reserverades för instansen, som vi pekade en domän mot. För att kunna köra via HTTPS öppnade vi därefter port 443 i brandväggen och använde Caddy som reverse proxy mot port 8080. Caddy ordnade automatiskt ett certifikat via Let's Encrypt.

Om ingenting annat anges skapar Label Studio en lokal sqlite-databas vid uppstart, för att spara information om användare och taggningsprojekt. Denna databas har vi tagit en backup på efter projektets slut för att ingen data ska gå förlorad.

Via gränssnittet går det att skapa en speciell länk för att bjuda in medarbetare till projektet. Via en miljövariabel går det att ange att denna länk ska vara enda möjligheten att skapa ett konto (se utökat docker-kommando nedan).

Inom docker-avbilden sparas data i mappen /label-studio/data. Denna vill vi länka till en lokal mapp.

Label Studio erbjuder ett antal möjligheter att inkludera den data som ska taggas. För våra syften, där vi har ett större antal bildfiler som inte är publikt åtkomliga och som ska taggas i omgångar, där samma bild kan förekomma i flera olika omgångar, valde vi att lägga bilddatan lokalt på instansen och länka till den så att den är tillgänglig internt i Label Studio (se även stycket om import / export nedan).

Ovanstående val ledde till att Label Studio kördes enligt följande:

docker run -it -p 8080:8080 \

-v `pwd`/data:/label-studio/data \

-v /home/username/byggrhandlingar\_img:/label-studio/data/kba/byggrhandlingar\_img \

--env LABEL\_STUDIO\_LOCAL\_FILES\_SERVING\_ENABLED=true \

--env LABEL\_STUDIO\_LOCAL\_FILES\_DOCUMENT\_ROOT=/label-studio/data/kba \

--env LABEL\_STUDIO\_DISABLE\_SIGNUP\_WITHOUT\_LINK=true \

heartexlabs/label-studio:latest

I vårt fall låg alltså bilddatan i mappen /home/<username>/byggrhandlingar\_img.

# Tillvägagångssätt för taggning

Det är viktigt att taggningen sker på ett konsekvent sätt, så att modellen kan lära sig av datan och inte lär sig olika saker beroende på vem som har taggat datan. Till exempel, vid riktningsangivelsen "Fasad mot norr", ska enbart ordet "norr" markeras eller alla tre ord? Därför tog vi fram ett dokument med instruktioner för hur taggningen skulle gå till. Därefter följde en testrunda, där vi taggade ett antal ritningar för att se om instruktionerna var tydliga och om det fanns några oklarheter. Efter testrundan justerades instruktionerna något.

Ett ytterligare sätt att säkerställa att taggningen skett på ett konsistent sätt är att låta det bli ett litet överlapp, så att flera annotatörer oberoenda av varandra taggar samma ritningar. Om de är överens om hur objekten ska taggas, är det troligt att resultatet är korrekt. Om de inte är överens, kan det vara ett tecken på att instruktionerna inte är tillräckligt tydliga.

Eftersom de olika typerna av bygglovshandlingar är ganska olika varandra och vi behövde träna flera modeller som var bra på att hantera de olika objekten vi letade efter, valde vi att dela upp taggningen i flera olika delprojekt. På så sätt kunde vi anpassa inflödet av data efter vad vi för tillfället behövde mer av. Exempel på två olika delprojekt var att markera skalstock och kompassriktning i situationsplaner, samt att markera avstånd till tomtgräns i situationsplaner och nybyggnadskartor.

Vi lade ofta upp flera projekt samtidigt, så att olika annotatörer kunde arbeta parallellt i sin takt. Vi använde en enkel färgkodning för att signalera vilka projekt som var pågående och vilka som var klara.

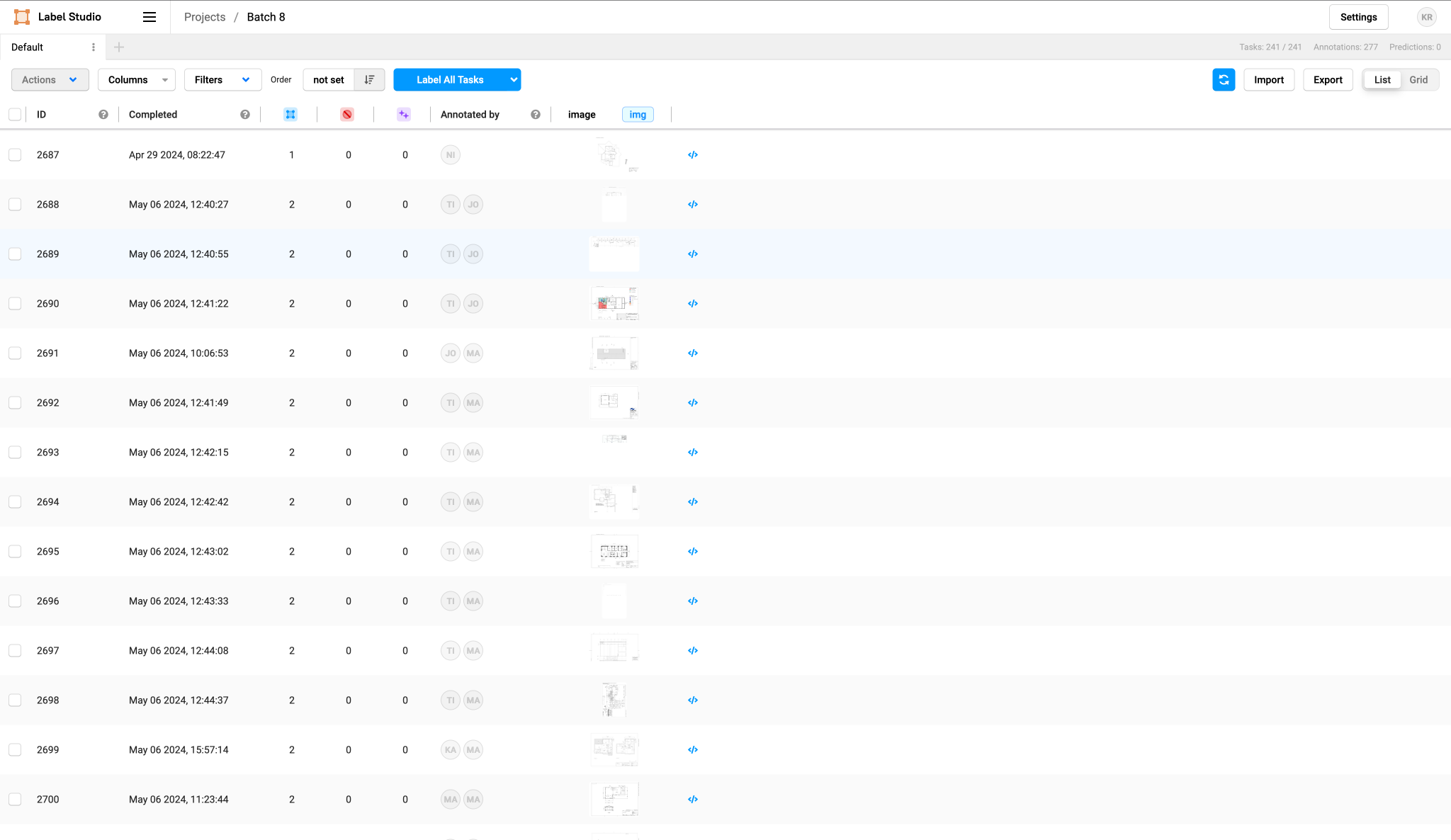


Bild: Ett delprojekt, som visar listan av bygglovshandlingar som ska taggas och vilka annotatörer som hanterat de olika handlingarna.



Bild: Färgkodade projekt i Label Studio

Olika objekt kräver olika typ av annotering. Skalstock och kompassriktning är enklast, då räcker det att rita ut en rektangel som täcker objektet. För att markera tomrgränser användes ett polygonverktyg, som kan användas för att rita ut oregelbundna former. För att markera marklinjer, som ofta ligger tätt intill varandra, användes punkter. Här fanns en av styrkorna med Label Studio, som hade inbyggt stöd för alla dessa verktyg. Det gick däremot inte enkelt att åstadkomma detta genom att klicka på en projektmall i gränssnittet, då samtliga projektmallar antog att användaren ville hålla sig till en typ av verktyg, till exempel bara rektanglar. Det gick dock enkelt att skapa sina egna projektmallar genom att kopiera den XML-kod som genererades för de olika verktygen och sätta ihop sin egen kombination.

Vi lade även till möjligheten att signalera att något av objekten finns, men är felaktigt, eller inte finns alls. Det visade sig även finnas ett behov att kunna markera att det var fel på hela bygglovshandlingen eller att annotatören vill ta upp ritningen för diskussion med hela gruppen.

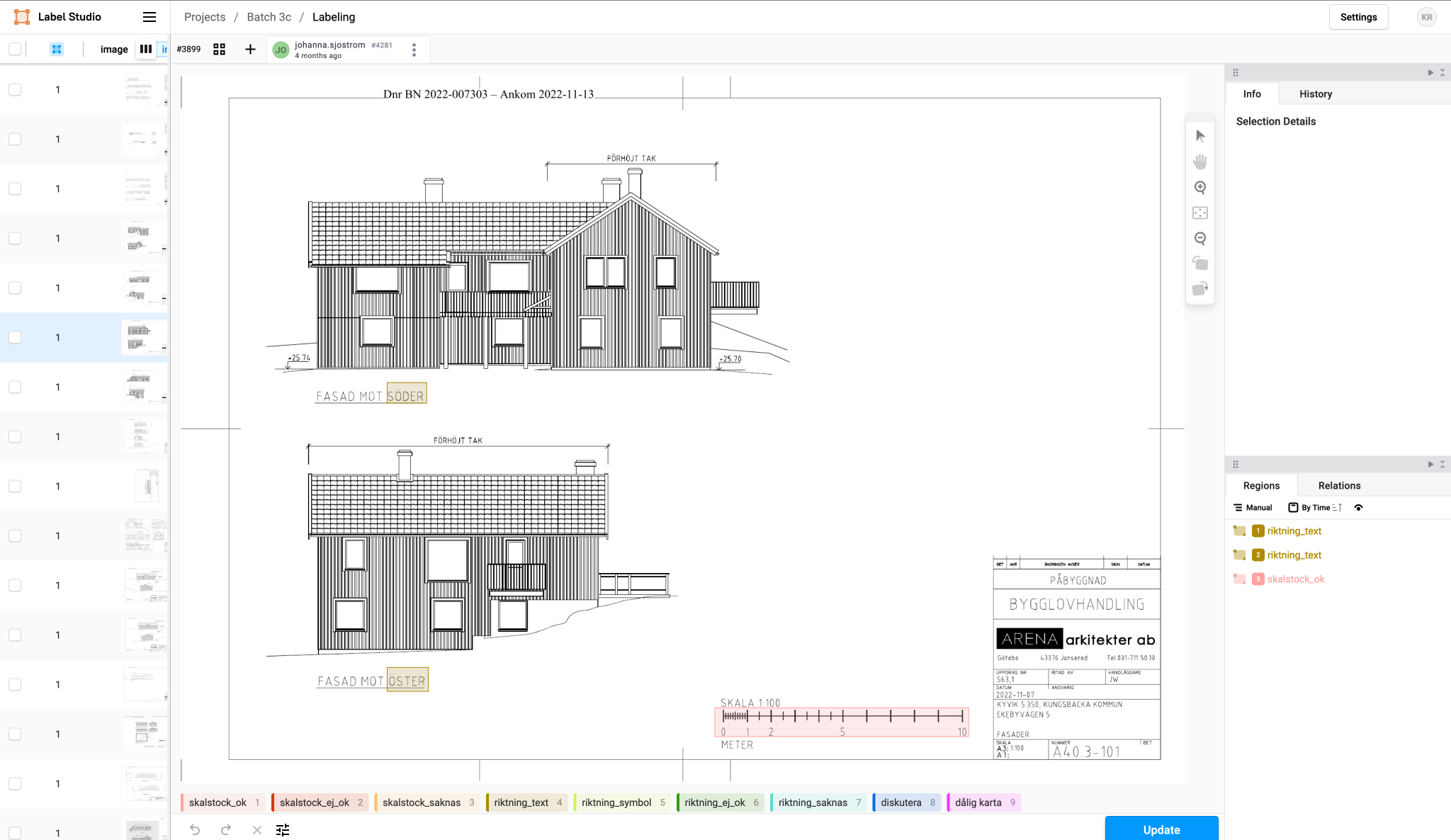


Bild: Kompassriktning och skalstock markeras med rektanglar i olika färg. Det går även att markera att något saknas eller att det är fel på ritningen.

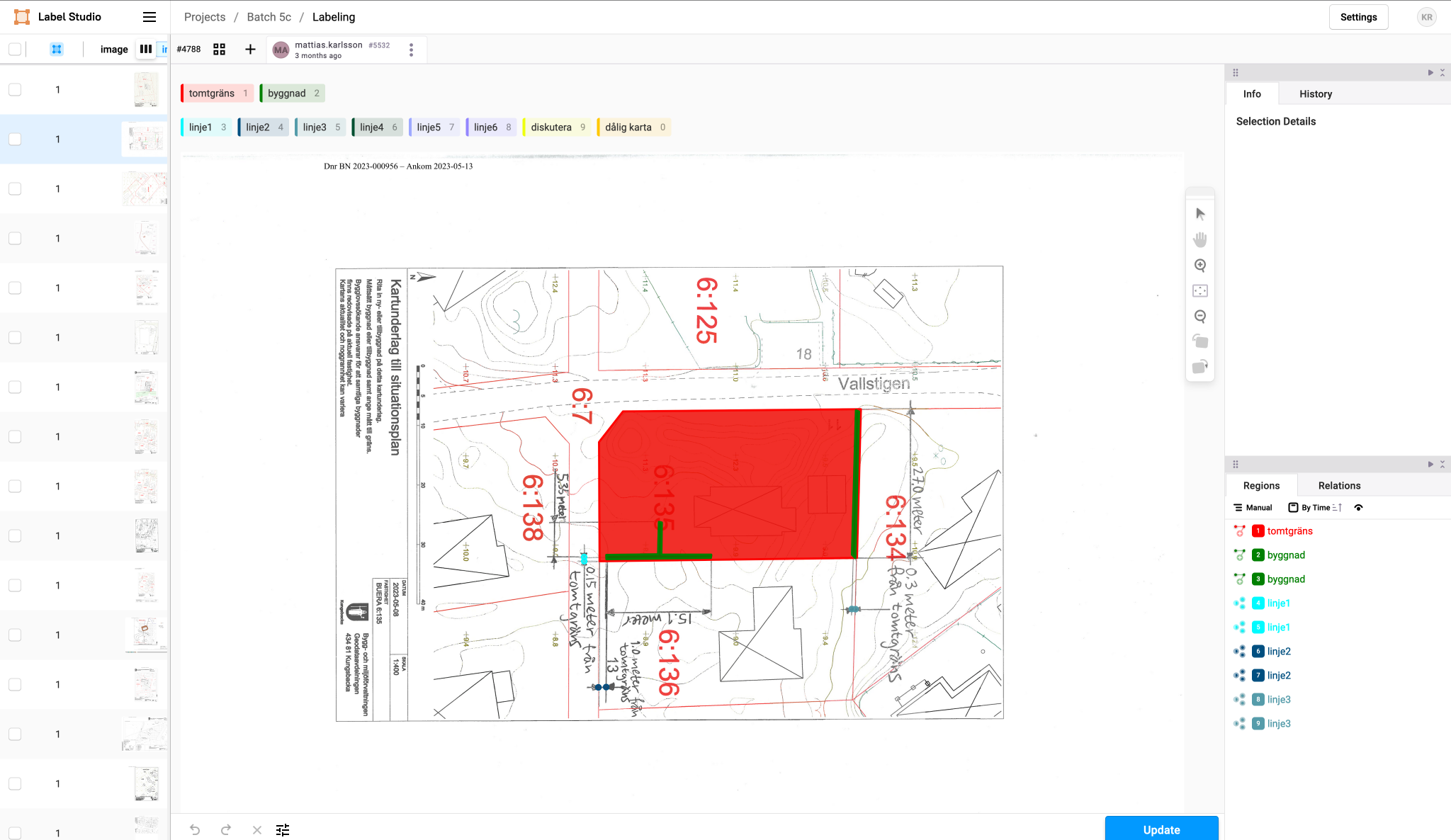


Bild: En annan uppsättning verktyg för tomter och avstånd till tomtgränser.

# Import / export

Som tidigare nämnts lade vi alla bygglovshandlingar lokalt på instansen. När ett nytt delprojekt skapades genererade vi en enkel JSON-fil som innehöll en lista på den delmängd av bildfiler som skulle taggas i just detta delprojekt. Själva importen av denna fil behövde sedan göras i en specifik ordning:

1. Skapa ett nytt projekt i Label Studio, men skippa dataimporten i detta steg. Ange i stället bara titel och färgkodning.

2. Gå in på nytt på projektets inställningar, nu finns en ny tabb “Cloud Storage”.

3. Lägg till en lokal mapp /label-studio/data/kba/byggrhandlingar\_img/byggrhandlingar, eller motsvarande som länkats i docker-kommandot. Kryssa inte i alternativet "Treat every bucket object as a source file" och ange inte något filnamnsfilter.

4. Importera JSON-filen med import-knappen ute i projektets huvudvy.

Exempel på JSON-fil:

[

{

"id": "965893",

"data": {

"image": "data/local-files/?d=byggrhandlingar\_img/byggrhandlingar/BN 2022-008022/Fasadritning/965893.png"

}

},

{

"id": "964747",

"data": {

"image": "data/local-files/?d=byggrhandlingar\_img/byggrhandlingar/BN 2022-007987/Fasadritning/964747.png"

}

}

]