Classify a photo of a property into a particular type

nested

Chris Brake

Automated image classification can save time for valuation agents

- Valuation is based on comparable properties listed for sale online
- Finding the ‘right’ images to compare takes time
- Can we automatically classify images to display the one(s) we need?
- Yes we can - with > 90% accuracy

The data?

- 20,000 images scraped from Zoopla, Rightmove
- Labelled via Amazon Mechanical Turk
- That look like this …...

Sample images

The Approach

- ‘Re-train’ the InceptionV3 (Google) CNN:
  - Transfer learning
  - Fine tuning
- Run on Google Virtual Machine:
  - 16 core 104GB
  - With optimised Tensorflow & Keras
- Workflow:

Process / Review data → Run models → Use model for predictions → Evaluation

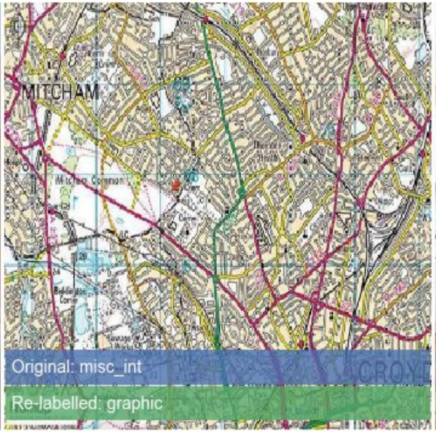
Key activity / code / learning ….

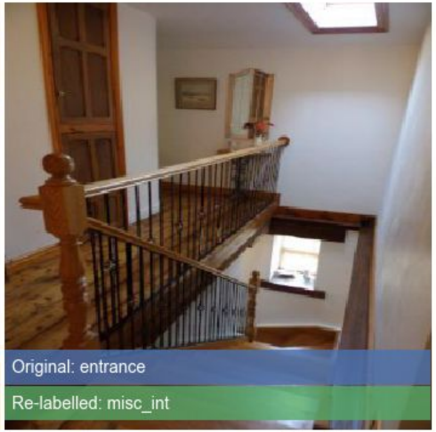
- Setting up a Google VM for data science:
  - See blog ….
  - Optimising Tensorflow installation matters (!)
  - How to move working files / directories on & off VM?
- Working with big image libraries:
  - Cataloguing - with recursive walks & Pandas DF
  - Review workflow (label correction)
  - Splitting train / val / test
- Image (grid) plotting:
  - Grids from df (variable size)
  - Labelling functions
  - Saving .png & .csv
- Labelling test data using model
- Keras (/Tensorflow):
  - Reconfigure top layer
  - Transfer learning & fine tuning
  - Loss curves
  - Callbacks are key

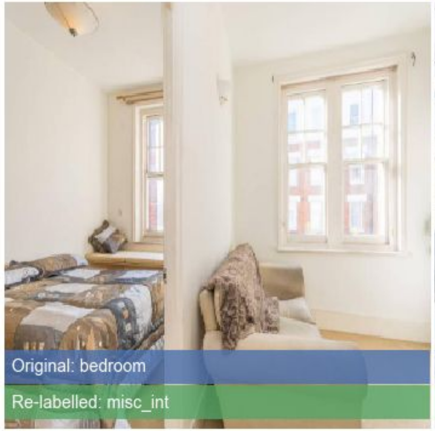
LOTS of material / code online on Keras transfer learning. Less on the rest …..

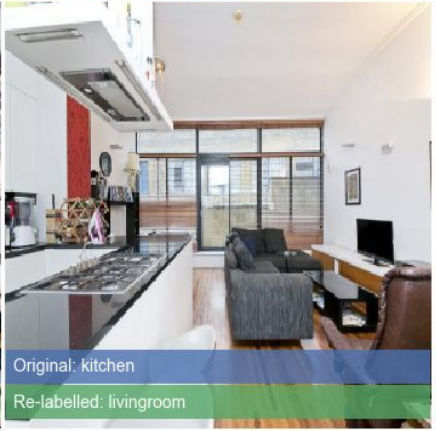
Some data challenges

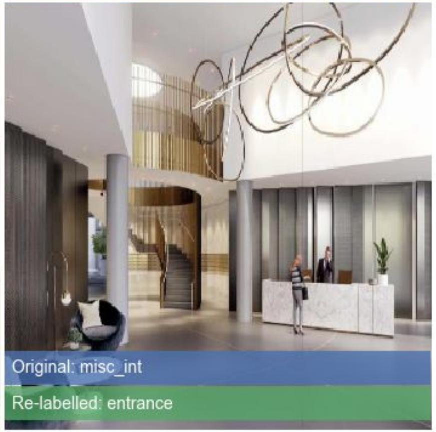
Reclassified images



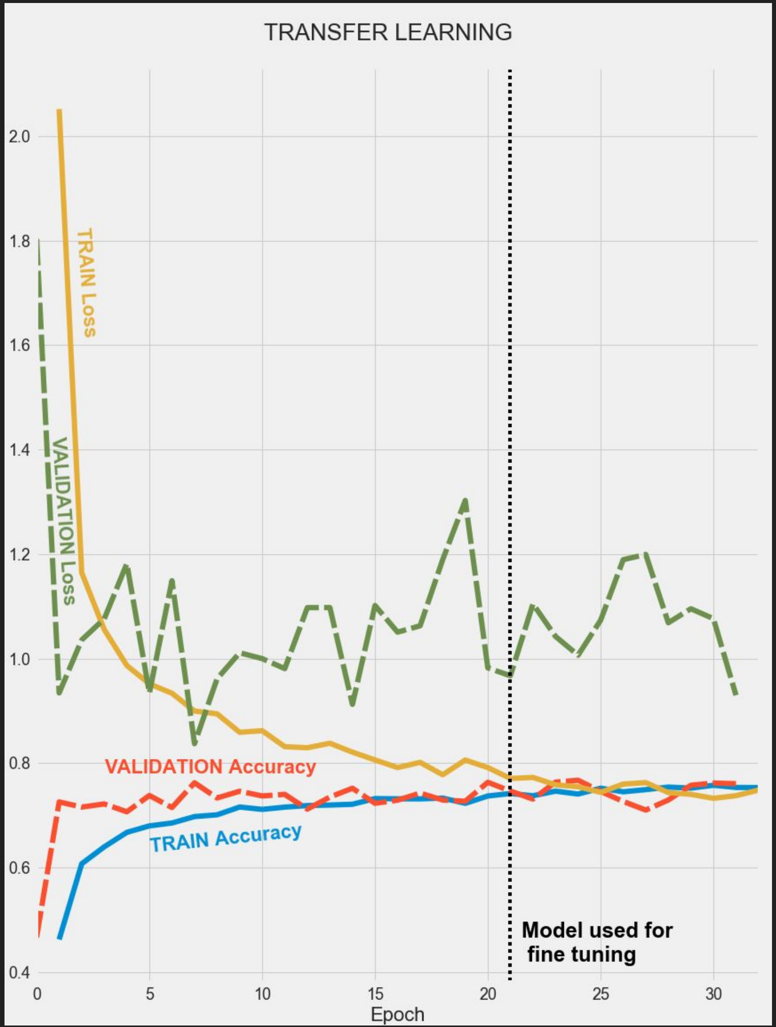




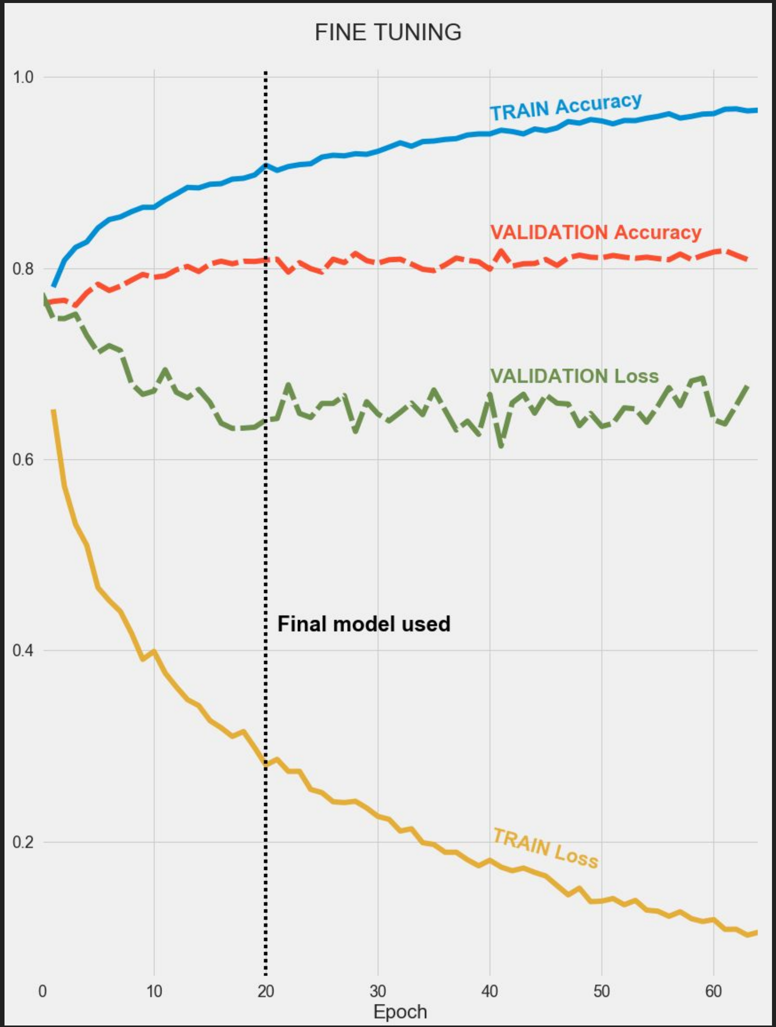




The training

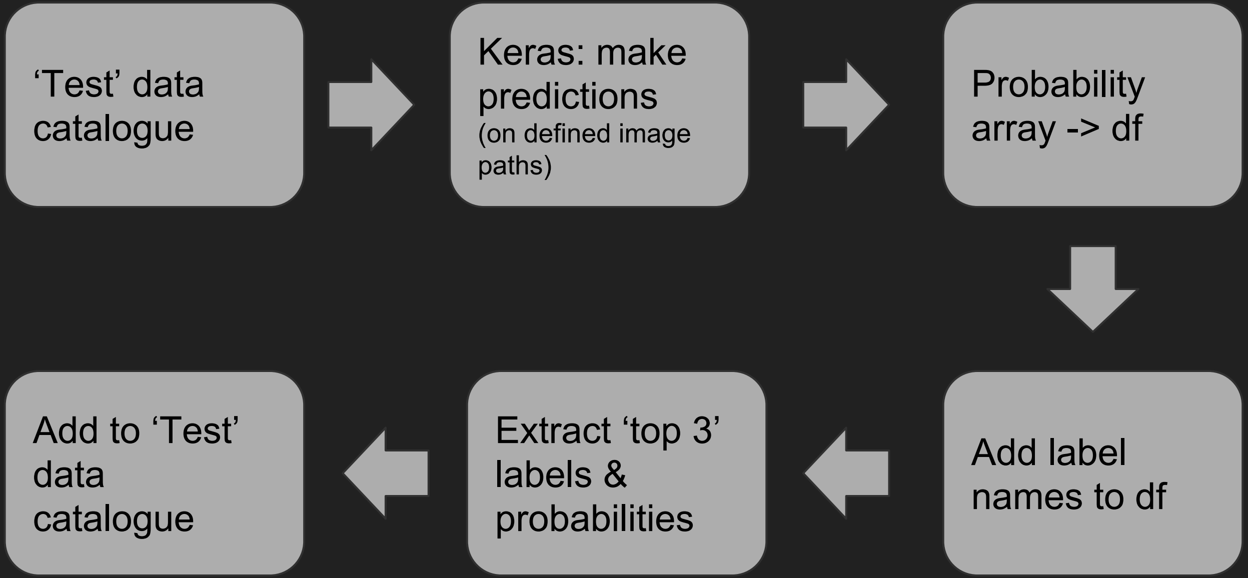


TRANSFER LEARNING
TRAIN Loss
VALIDATION Loss
VALIDATION Accuracy
TRAIN Accuracy
Model used for fine tuning
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
5
10
15
20
25
30
Epoch

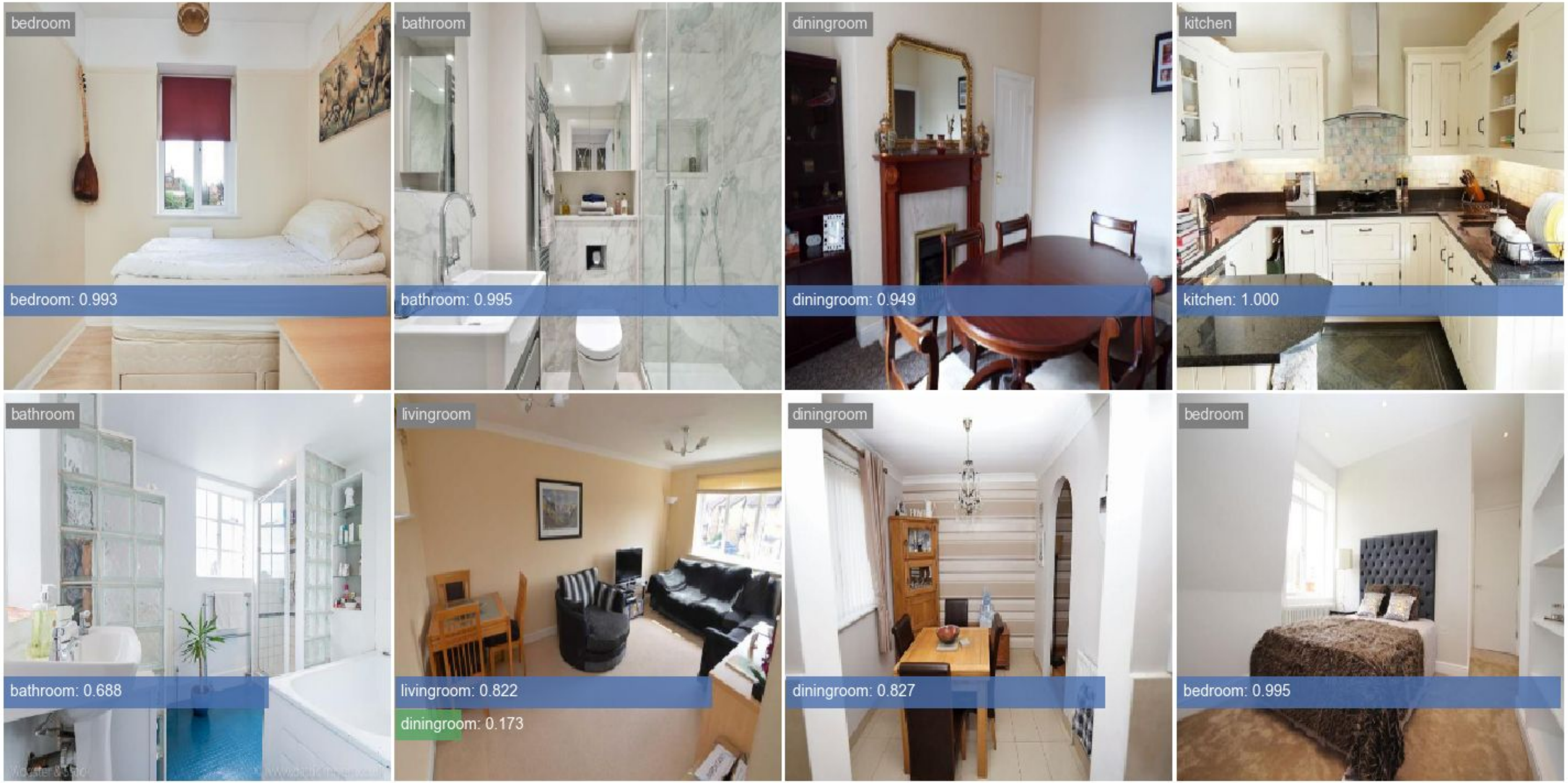


FINE TUNING
TRAIN Accuracy
VALIDATION Accuracy
VALIDATION Loss
Final model used
TRAIN Loss
1.0
0.8
0.6
0.4
0.2
0
10
20
30
40
50
60
Epoch

Making Predictions



Label threshold @ 10% for correctly labelled images



Evaluation

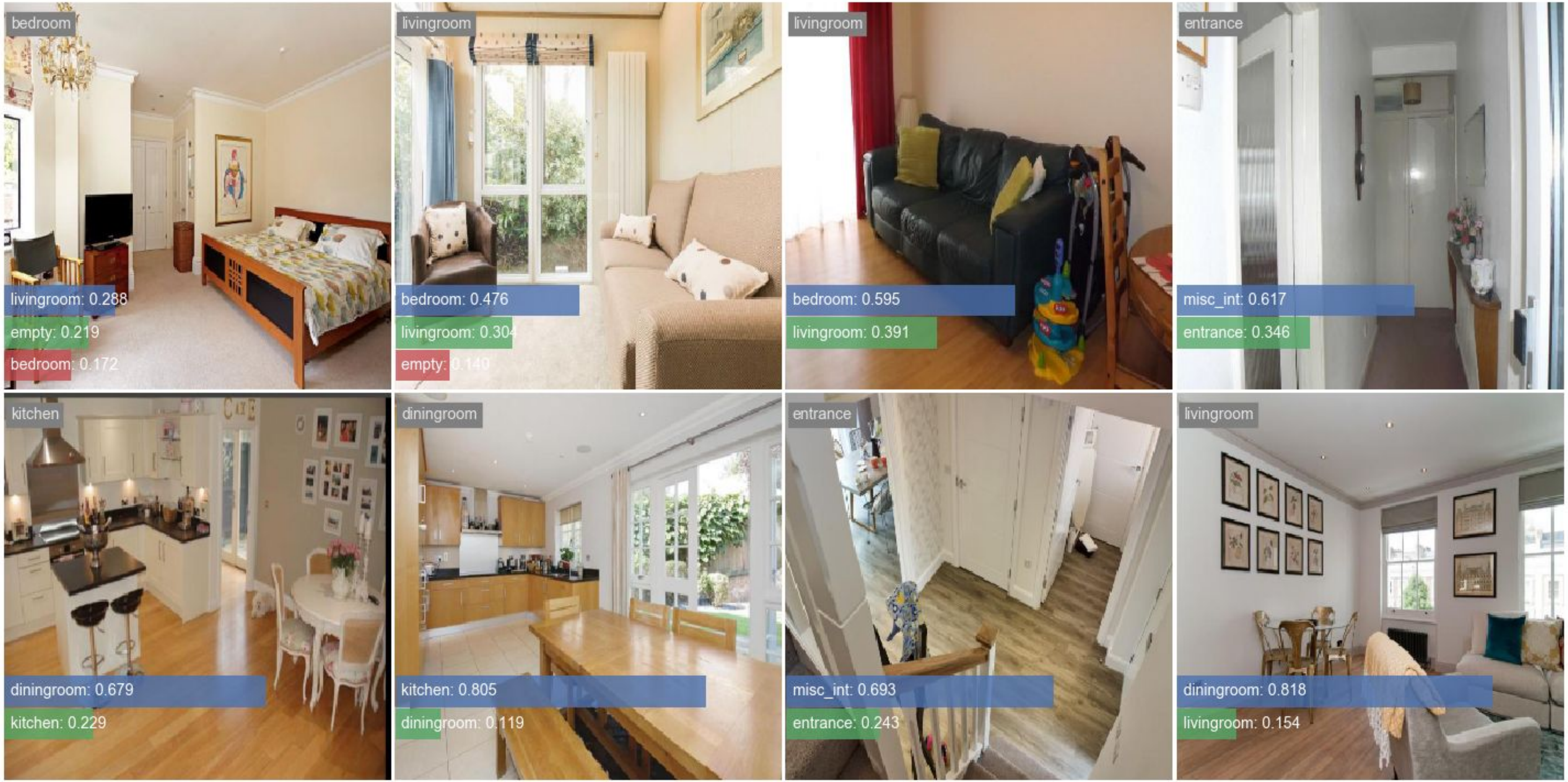
Accuracy = 85%

BUT

>90% for key rooms

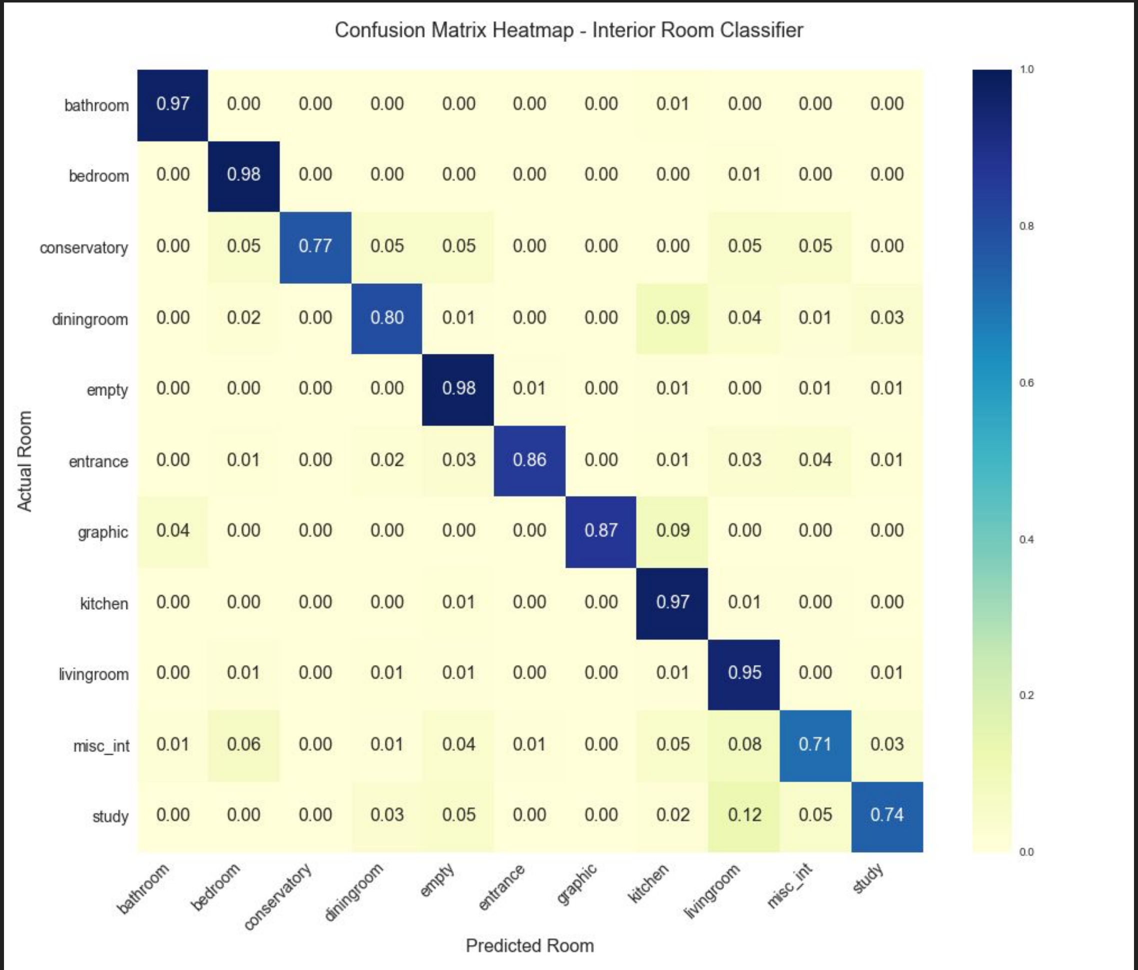
And what about the ambiguity / multiple potential labels problem?

Label threshold @ 10% for 'mislabelled' images



Consider top 3 labels (if >10%):

Accuracy = 93.8%



Further work

- Revisit image data labelling:
    - Reduce categories / reduce granularity (e.g. misc_int & entrance)
    - More samples on some categories (e.g. floorplans)
    - Multiple voting per image, and multiple labels (primary / secondary)

- Look at classifying internal AND external images in single model
    - Think this will work well once image labelling revisited