Multi-class Image Recognition of Food

Michael Mortensen (s144031), Robert Fanning (s172419), Manlu Xu (s161407), Huayu Zheng (s162077), and Ting Ding (s172016)

DTU Compute · Technical University of Denmark

1 Goals

- Image classification for determining nutritional content for calculating insulin dosage.
- Acquisition and cleaning of large (approx. 10,000) food image dataset.
- Annotation of food images for Faster R-CNN.
- Simple high performance CNN for classifying single labeled images.
- Faster R-CNN combining both RPN and Fast R-CNN for multi-labeled images.

2 Data acquisition

Food image dataset acquired using image scraping tool. Download from several image hosting websites (e.g Flikr, Google, Bing etc.). Dataset was processed and manually cleaned of any irrelevant images.

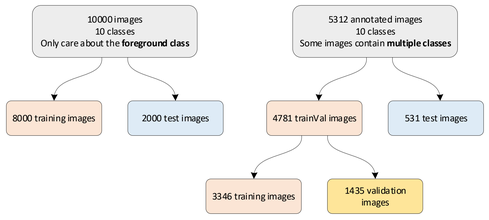


Figure 2: Dataset structure, left Simple CNN, right Faster R-CNN

* 10 classes: glass of milk, glass of water, plain rice, plain spaghetti, slice of bread, boiled peas, boiled potatoes, chopped lettuce, fried egg, meatballs

3 Simple Convolutional Neural Network

3.1 Architecture

Parameters	Conv1	Max1	Conv2	Max2	Conv3	Max3	FC1	FC2
in channels	3		64		128			
out channels	64		128		512			
kernelsize	3	2	4	2	4	2		
stride	2	2	2	2	2	2		
out dimension	149		73		35			
in features							8192	2000
out features							2000	200

TABLE 1: CNN model parameters.

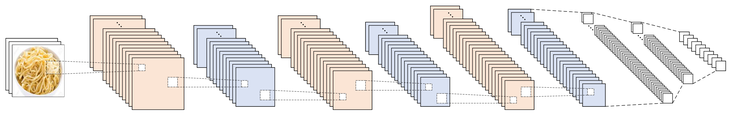


Figure 3: Simple CNN Architecture

Activation:

- The rectified linear unit (ReLU) activation function defined by $f(z) = max(0, z)$.
- Sparse activation, better gradient propagation, and efficient computation.

Regularization:

- Batch normalization and L2 regularization (weight decay).

4 Faster Region Proposal Convolutional Neural Network (Faster R-CNN)

4.1 Architecture

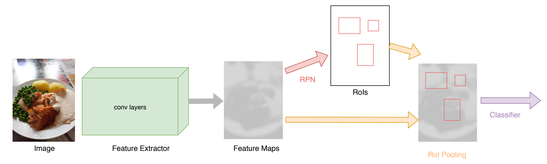


Figure 4: Faster R-CNN Architecture

- Feature Extractor

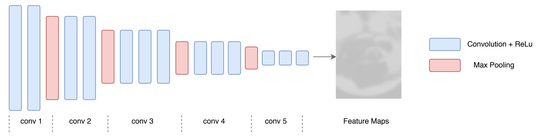


Figure 5: VGG16 model

- Region Proposal Network

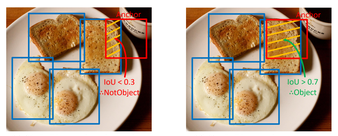
Figure 6: Region Proposal Network

- RoI Pooling
  RPN outputs multiple regions of interest with varying sizes to be used by the Fast R-CNN. Region of Interest Pooling then solves the variation of sizes by scaling the feature maps to the same size (e.g 7x7) enabling the reuse of the feature map.

4.2 Loss Function of Region Proposal Network

Intersection over Union

$$IoU = \frac{Anchor \bigcap Ground-truthBox}{Anchor \bigcup Ground-truthBox} \begin{cases} Object & > 0.7 \\ NotObject & < 0.3 \end{cases} \quad (1)$$



((a)) IoU overlap less than 0.7

((b)) IoU overlap greater than 0.7

Figure 7: Loss Function as computed using Intersection-over-Union

If 0.3<IoU<0.7 it does not contribute to the training.

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{cls}} \sum p_i^* L_{reg}(t_i, t_i^*) \quad (2)$$

- p_i: Predicted probability of anchor i being an object.
- p_i^*: Ground-truth label (1 if anchor is an object, 0 if not an object)
- t_i: Vector containing 4 coordinates of the predicted bounding box.
- p_i^*: Vector containing 4 coordinates of the Ground-truth box.

4.3 Training RPNs

- End-to-end training using back-propagation and stochastic gradient descent.
- Loss function computed using 256 randomly samples from a mini-batch
- Layers weights randomly initialized using zero-mean Gaussian distribution with 0.01 standard deviation
- Learning rate = 0.001, Momentum = 0.9, Weight Decay = 0.0005

4.4 Step Alternating Training: RPN and Fast R-CNN

Training scheme is an alternating 4 step algorithm

- STEP 1: Train RPN for region proposal task.
- STEP 2: Use proposals from RPN to train Fast R-CNN (Detection Network).
- STEP 3: Fast R-CNN is then used to initialize RPN. Only layers unique to RPN are tuned, shared convolutional layers are fixed.
- STEP 4: Again with shared convolutional layers fixed, the unique layers of the Fast R-CNN are tuned.

4.5 Implementation details

p2.xlarge instance of AWS; caffe-pretrained VGG16 pretrain model; Pytorch frameware

5 Experiments

5.1 Simple CNN

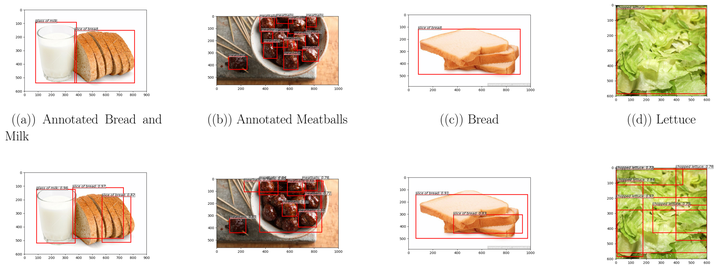
Training parameters and results using 2158 test images:

Parameter	Value
Epochs	8
Batch size	512

Class	Test accuracy
All	60.00%
Plain spaghetti	51.76%
Slice of bread	52.08%
Fried egg	65.73%
chopped lettuce	65.33%

Class	Test accuracy
Glass of milk	64.44%
Boiled potatoes	58.29%
Boiled peas	51.00%
Plain rice	51.26%
Meatballs	75.76%
Glass of water	67.98%

5.2 Faster R–CNN



((a)) Annotated Bread and Milk

((b)) Annotated Meatballs

((c)) Bread

((d)) Lettuce

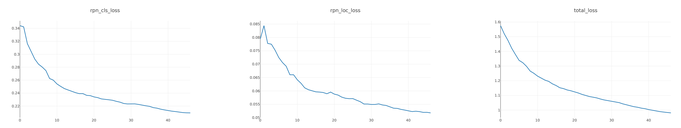
((e)) Test result 1

((f)) Test result 2

((g)) Test result 3

((h)) Test result 4

Figure 8: Annotated and predicted images with bounding box in testing



((a)) Classification loss in RPN

((b)) Regression loss in RPN

((c)) Total loss

Figure 9: Training loss

6 Conclusions & Outlook

- Simple CNN easy to train and efficient to implement, however fails for multi-labeled classification, high sensitivity to quality of dataset.
- Faster R-CNN, successfully enables mutli-labeled classification with high accuracy even with noisy dataset. Sensitive to annotation bias.