6.869 Mini-Places Challenge

Eduardo De Leon, Harini Suresh, Nicholas Locascio, Tiffany Wong

Scene Classification Task

- Goal: identify scene categories depicted in a photograph.
- Output: for each image, the top 5 scene categories
    - (1) locations can be multipurpose e.g. restaurant & bar
    - (2) humans use different words to describe the same place e.g. woods, forest

Problems with Traditional Scene Classification Model Training

- Assumption that scenes exist in orthogonal space is not true.

Prediction:
Bathroom

Prediction:
Volcano

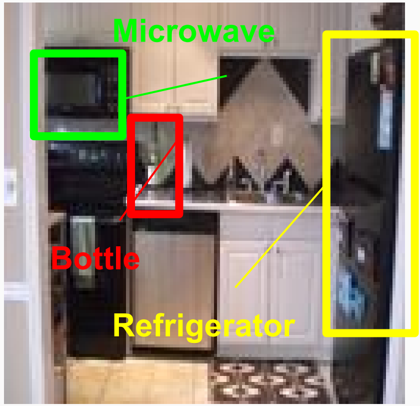
Same Loss!

True Label:
Shower

Ideally should penalize the model **more** for thinking it is a "Volcano"

Object Annotations

- We should penalize the model **more** for confusing scene to be Volcano.
- Dataset contains **object annotations** useful to this end.
- Can create **auxiliary tasks** to predict these object annotations
- **Auxiliary tasks** create more informative loss function
  - Results in smoother gradient



Related Works

- Training with auxiliary targets [1]
  - Train using auxiliary targets
  - Help disentangle confounding factors
  - Better characterize what distinguishes different scenes
  - Regularizing effect
- Utilizing label text embeddings [2]
  - Easier for the network to predict a semantic label embedding
  - Provides a way to incorporate additional information into the network

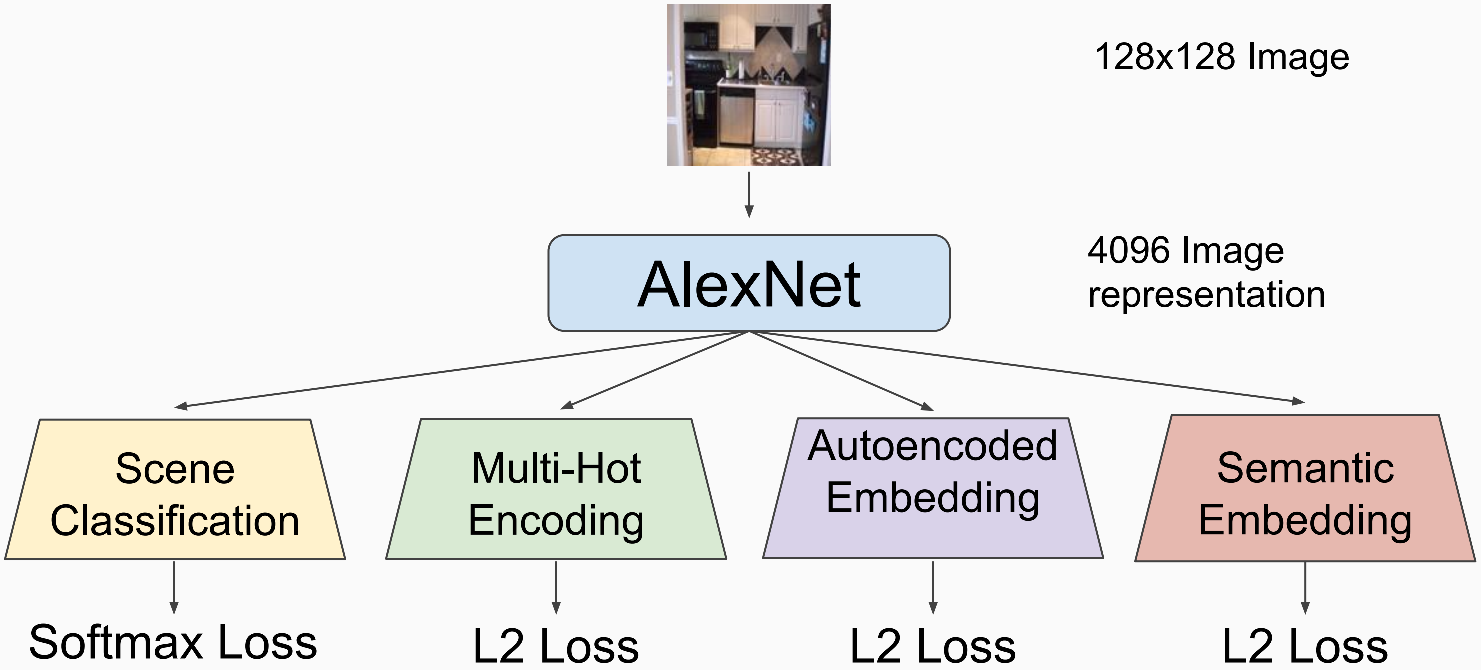
[1] Zhong & Ettinger 2016, [2] Akata et. al. 2015

Goal: Explore Effect of Object Embeddings on Scene Prediction

Choosing an Object Representation

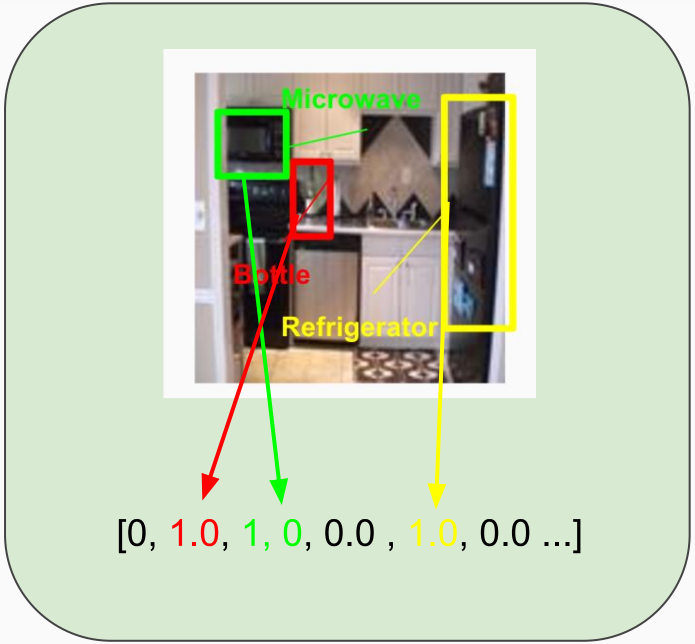
a. Multi-hot object encoding
b. Multi-Hot Object Autoencoder Embedding
c. Semantic Object Embedding via Word2Vec word vectors

Our Model


128x128 Image
AlexNet
4096 Image representation
Scene Classification
Multi-Hot Encoding
Autoencoded Embedding
Semantic Embedding
Softmax Loss
L2 Loss
L2 Loss
L2 Loss

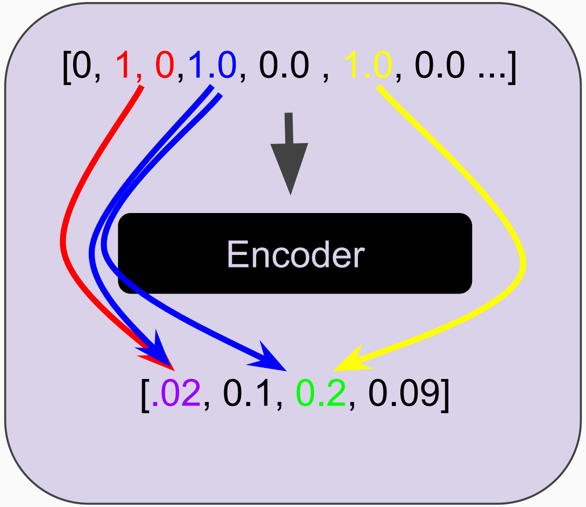
Multi-Hot Encodings as Auxiliary Task

- Each object has a category label
- Sum all category labels of objects in image to create **Multi-Hot Encoding**



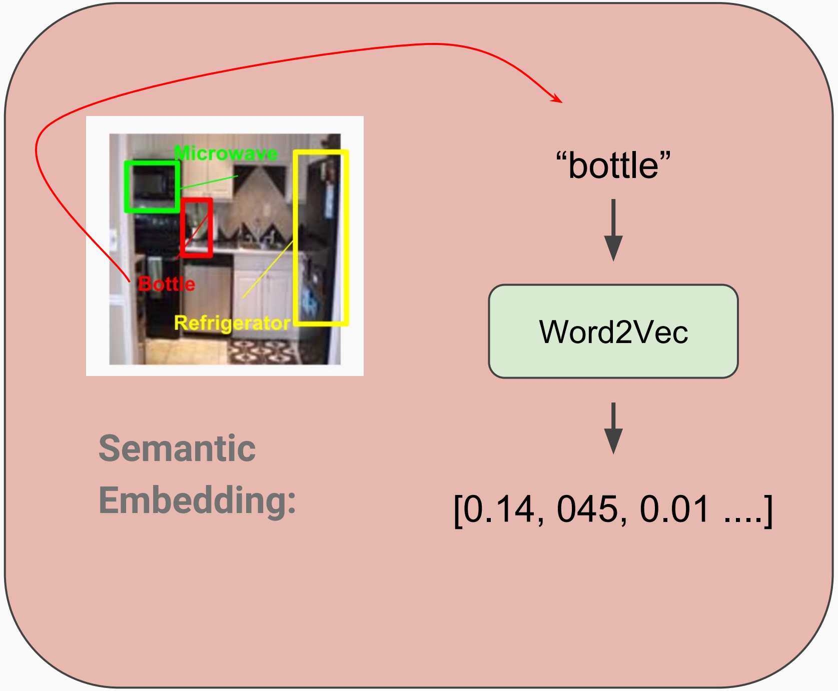
Auto-Encoded Object Embeddings as Auxiliary Task

- Train an autoencoder on training and validation Multihot-Encodings
- Pass each Multi-Hot Encoding through the encoder for a 40 dimensional **Relational Encoding** of the image objects



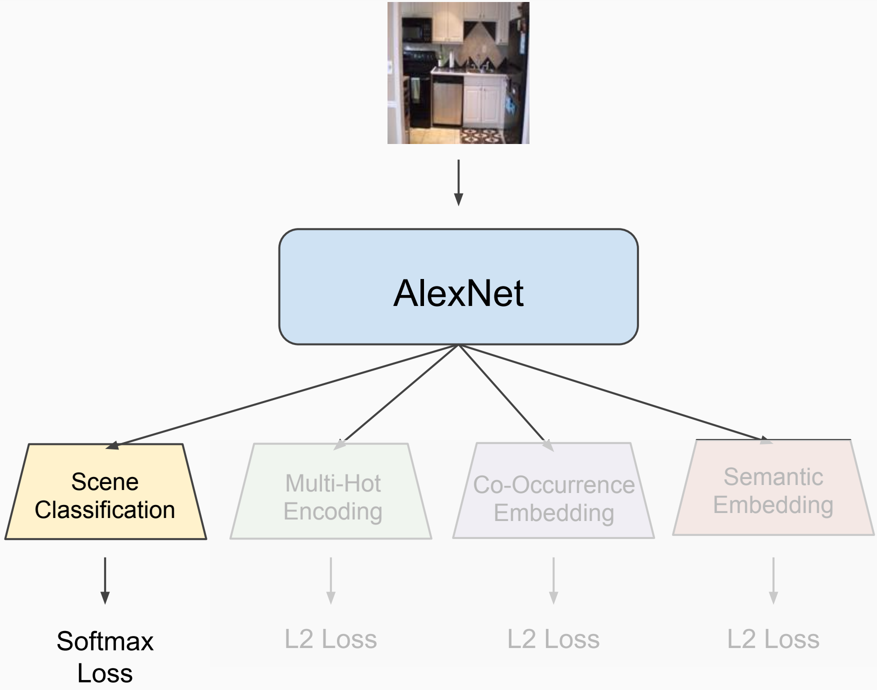
Semantic Embeddings as Auxiliary Task

- Each object has a string label
- For each object, get the string label, run word through pretrained **Word2Vec** model
    - Some object categories are multi-word, so for these we average all the words embeddings of an object together.
- Average all these together for all objects in an image to get **Semantic Embedding**



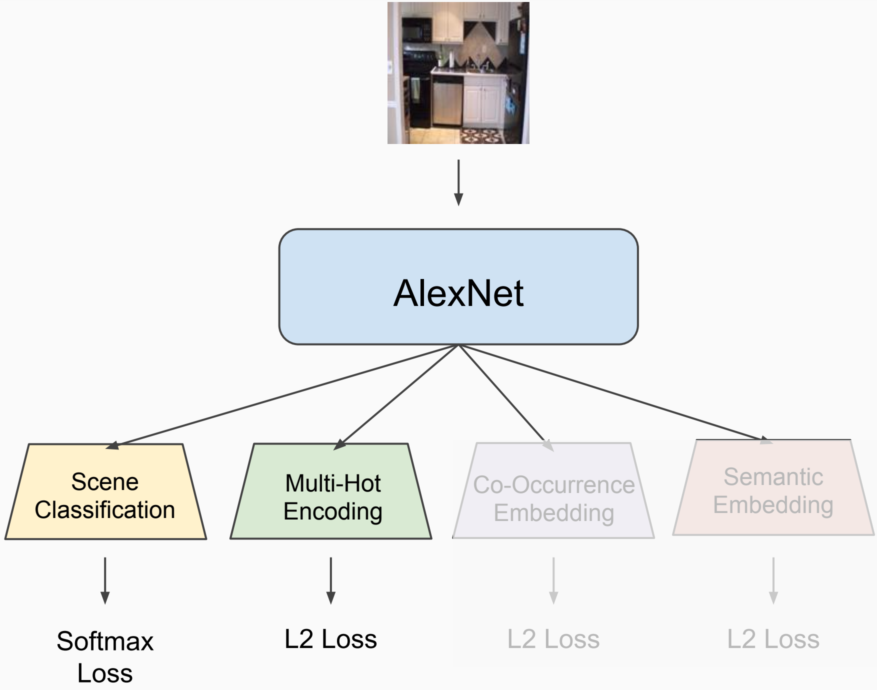
Experiments

- Run 5 different models
  - **Scene Classification Baseline**
  - Scene + Multihot Objects
  - Scene + Co-Occurence Embedding
  - Scene + Semantic Embedding
  - Scene + All Embeddings



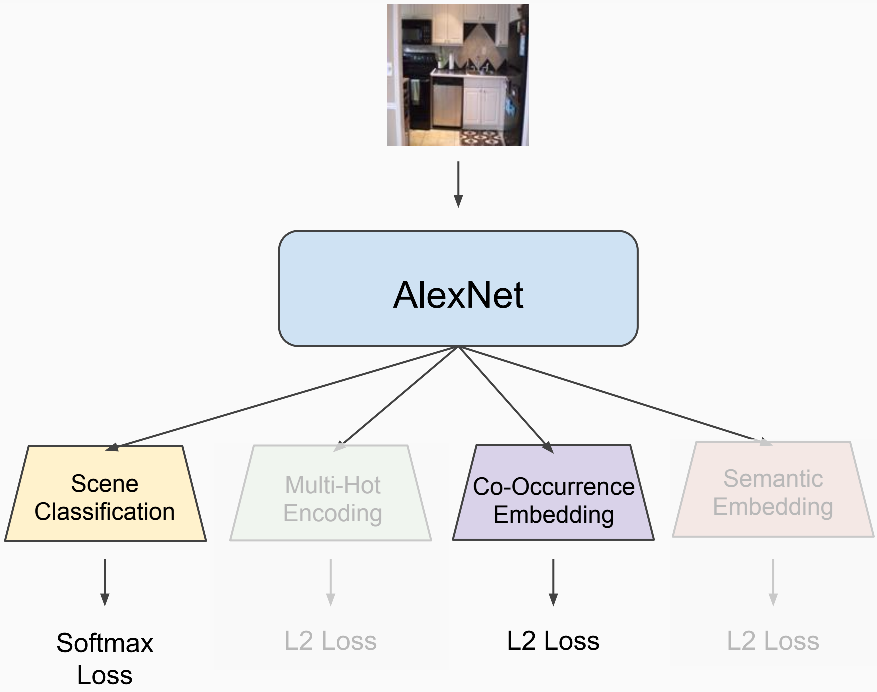
Experiments

- Run 5 different models
  - Scene Classification Baseline
  - **Scene + Multihot Objects**
  - Scene + Co-Occurrence Embedding
  - Scene + Semantic Embedding
  - Scene + All Embeddings



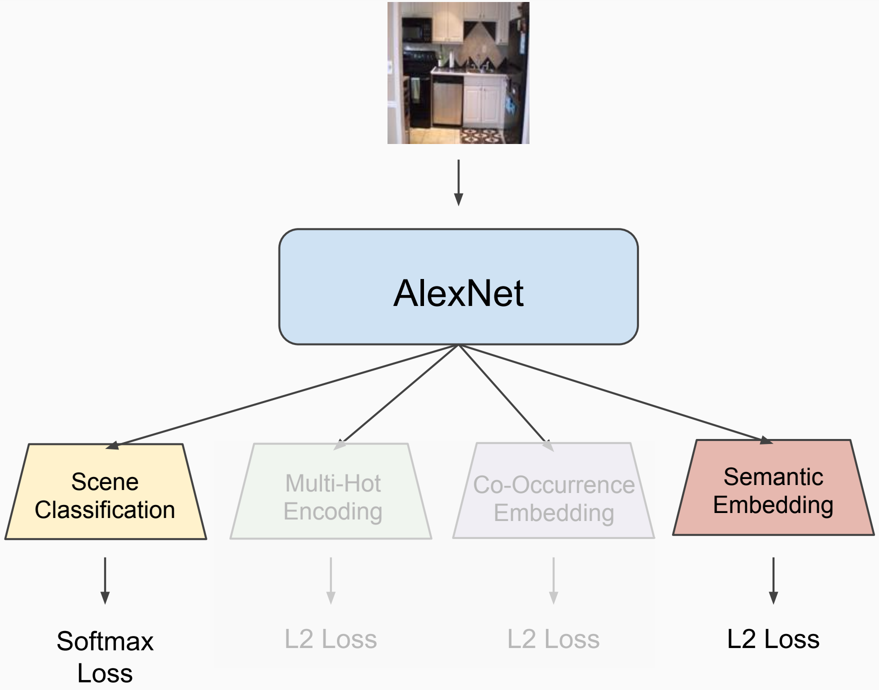
Experiments

- Run 5 different models
  - Scene Classification Baseline
  - Scene + Multihot Objects
  - **Scene + Co-Occurrence Embedding**
  - Scene + Semantic Embedding
  - Scene + All Embeddings



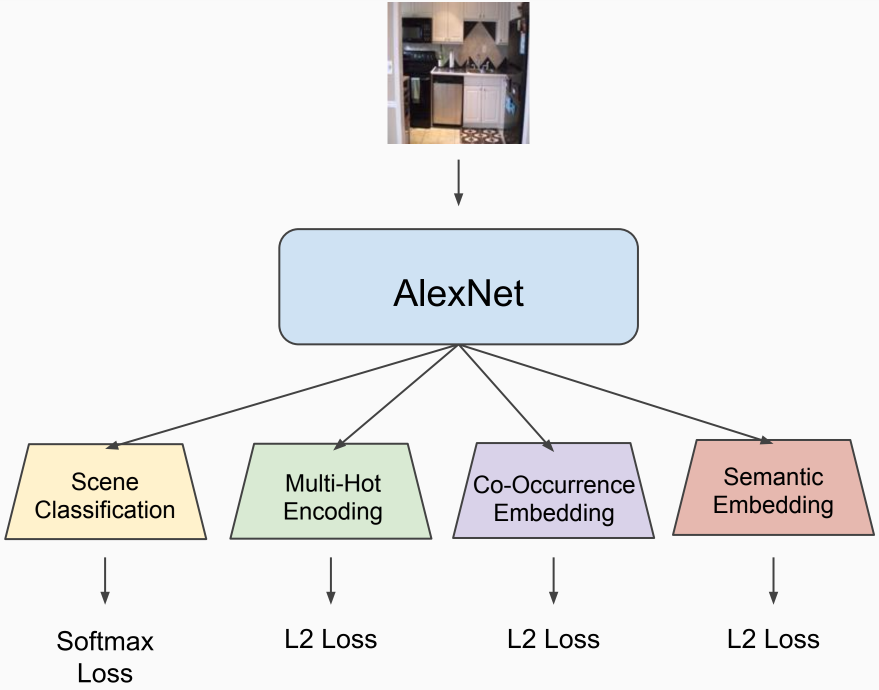
Experiments

- Run 5 different models
  - Scene Classification Baseline
  - Scene + Multihot Objects
  - Scene + Co-Occurrence Embedding
  - **Scene + Semantic Embedding**
  - Scene + All Embeddings

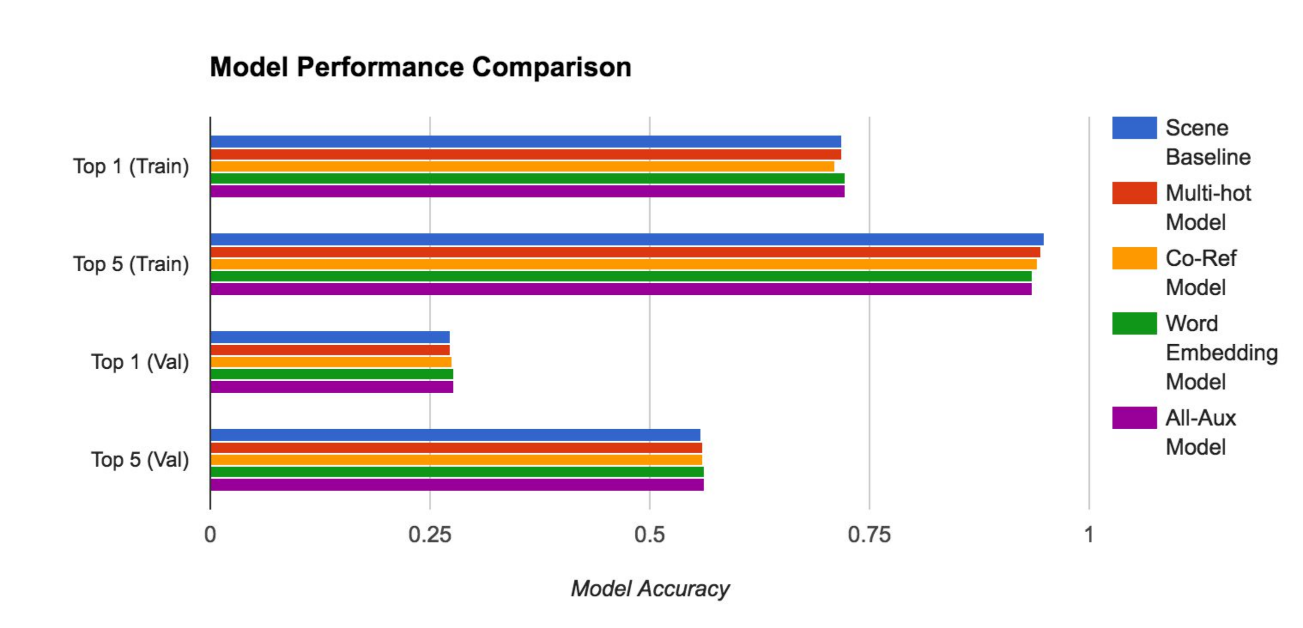


Experiments

- Run 5 different models
  - Scene Classification Baseline
  - Scene + Multihot Objects
  - Scene + Co-Occurence Embedding
  - Scene + Semantic Embedding
  - **Scene + All Embeddings**



Results



Model Performance Comparison
Top 1 (Train)
Top 5 (Train)
Top 1 (Val)
Top 5 (Val)
0
0.25
0.5
0.75
1
Model Accuracy
Scene Baseline
Multi-hot Model
Co-Ref Model
Word Embedding Model
All-Aux Model

Discussion

- We achieve best results using our full model (+0.46%)
- Marginal improvement over baseline - why so little?
- **Main Reason:** We only have object annotations for **3.5%** of training images.
  - This means that our gradient updates are the same for 96.5% of all training examples for all 5 models.
- **Future work:** Try techniques on a dataset that contains objects annotations for all images, not just 3.5%