Uncertainty modelling within an End-to-end framework for Food Image Analysis

Petia Radeva

Collaboration with:
Eduardo Aguilar, Marc Bolaños,
Bhalaji Nagarajan, Rupali Khatun

University of Barcelona &
Computer Vision Center

petia.ivanova@ub.edu
Contents

• The food image problem
• Multi-task food learning with aleatoric uncertainty
• Food recognition with epistemic uncertainty
• Conclusions
Why food recognition?

"Camera eats first"

180M #food
90/minute

54% take picture
39% post it
Why is the food recognition a challenge?
Motivation

Food Analysis Problems

- Intra-class variability
- Inter-class similarity

Decrease in Precision
Are we able to recognize thousands of dishes?

- 79% on UECFOOD
- 44% on ChinaFood1000
- How to achieve scalability?
Contents

• The food image problem

• Multi-task food learning with aleatoric uncertainty

• Food recognition with epistemic uncertainty

• Conclusions
Food Analysis as a Multi-task Problem

- Learning **multiple objectives** from a shared representation
  - Efficiency and prediction **accuracy**.

- Crucial importance in systems where **long computation** run-time is prohibitive
  - Combining all tasks **reduces computation**.

- Inductive **knowledge transfer**
  - **Generalization** by sharing the domain information between complimentary tasks.

**Cuisine:** French.

**Categories:** Meat.

**Ingredients:** salt, oil, onion, garlic, black pepper, tomato, cloves, parsley, thyme, bay, white wine, clove, duck, fat, mutton.

**Dish:** Confit de canard.
Transfer Learning

Fine-tuning

Multi-task learning

Cuisine: French.
Categories: Meat.
Ingredients: salt, oil, onion, garlic, black pepper, tomato, cloves, parsley, thyme, bay, white wine, clove, duck, fat, mutton.
Dish: Confit de canard.
Multi-task FAQ

• How should one pick the right architecture for multi-task learning?

• Does it depend on the final tasks?

• Should we have a completely shared representation between tasks?

• Or should we have a combination of shared and task-specific representations?

• Is there a principled way of answering these questions?
Food Recognition as a MTL

\[ L_{\text{total}} = \sum_{i} \omega_i L_i \]
How to define the importance of each task?

- Weighted uniformly the losses.
- Manually tuned the losses.
- Dynamic weighted of the losses.
  - The main task is fixed and weights are learned for each side-task ([1]).
  - Weight the tasks according to the homoscedastic uncertainty ([2]).

Let’s talk about uncertainty
But many unanswered questions...

- Why doesn’t my model work?
- -> Why does my model work?
- We don’t understand many of the tools that we use...
  - E.g. stochastic reg. techniques (dropout) are used in most deep learning models to avoid over-fitting. Why do they work?
- What does my model know?

From Gal’16
But many unanswered questions...

- Why does my model work?
- What does my model know?
- Why does my model predict this and not that?

- **Our models are black boxes and not interpretable...**
- Physicians and others need to understand why a model predicts an output.

Gal’16
Uncertainty in ML

• For Computer scientists, computers and algorithms are deterministic.

“Many branches of computer science deal mostly with entities that are entirely deterministic and certain. Given that many computer scientists work in a relatively clean and certain environment, it can be surprising that machine learning makes heavy use of probability theory.”

• The reason that the answers are unknown is because of uncertainty.

• The solution is to systematically evaluate different solutions until a good or good-enough set of features and/or algorithm is discovered for a specific prediction problem.

Noise in observations

- Noise refers to *variability or randomness* in the observation.

- The real world, and in turn, real data, is *messy or imperfect*.
  - As practitioners, we must remain skeptical of the data and develop systems to expect and even harness this uncertainty.

Incomplete Coverage of the Domain

• In statistics, a random sample refers to a collection of observations chosen from the domain without systematic bias.
  o However, there will always be some bias.

• A suitable level of variance and bias in the sample is required such that the sample is representative of the task or project for which the data or model will be used.
  o Often, we have little control over the sampling process.

Incomplete Coverage of the Domain

- In all cases, we will never have all of the observations. If we did, a predictive model would not be required.

- This is why we **split a dataset into train and test sets** or use resampling methods like k-fold cross-validation.
  
  - We do this to handle the uncertainty in the representativeness of our dataset and estimate the performance of a modelling procedure on data not used in that procedure.

Imperfect Model of the Problem

- This is often summarized as “all models are wrong,” or more completely in an aphorism by George Box:

  “All models are wrong but some are useful”

- This does not apply just to the model, the artifact, but the whole procedure used to prepare it, including the choice and preparation of data, choice of training hyperparameters, and the interpretation of model predictions.

Imperfect Model of the Problem

- Another type of error is an error of omission.

“In many cases, it is more practical to use a simple but uncertain rule rather than a complex but certain one, even if the true rule is deterministic and our modeling system has the fidelity to accommodate a complex rule.”

- Given we know that the models will make errors, we handle this uncertainty by seeking a model that is good enough.
  - This often is interpreted as selecting a model that is skillful as compared to a naive method or other established learning models, e.g. good relative performance.

How to manage Uncertainty

- Probability is the field of mathematics designed to handle, manipulate, and harness uncertainty.

- **In terms of noisy observations**, probability and statistics help us to understand and quantify the expected value, the variability of variables in our observations from the domain.

- **In terms of the incomplete coverage of the domain**, probability helps to understand and quantify the expected distribution and density of observations in the domain.

- **In terms of model error**, probability helps to understand and quantify the expected capability and variance in performance of our predictive models when applied to new data.

Why uncertainty is important?

Why uncertainty is important?
1. Given a model trained with several pictures of fruits, a user asks the model to decide what is the object using a photo of a chocolate cake.

Who is the guilty for this?

\[ S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \]
2. We have different types of images to classify fruits, where one of the category comes with a lot of clutter/noise/occlusions.

Adapted from Gal (2016)
Model uncertainty

3. What is the best model parameters that best explain a given dataset? What model structure should we use?

Gal (2016)
Model uncertainty

Noisy labels: with supervised learning we use labels to train the models.

If the labels are noisy, the uncertainty increases.
Types of uncertainty in Bayesian modeling

**Aleatoric** – captures the noise inherent in the observations

- heteroscedastic – data-dependent
- homoscedastic – constant for different data points, but can be task-dependent.

**Epistemic** – model uncertainty

- Can be explained away given enough data
- Uncertainty about the model parameters
- Uncertainty about the model structure
How to determine the total loss of the MTF?
- Expensive to learn & Affects the performance and the efficiency.

Use aleatoric uncertainty modeling to make the model smarter!

$$L_{\text{total}} = \sum_{i} \omega_i L_i$$
Let us consider a neural network defined on T tasks with model output y and parameters W. Factorizing the output and assuming a Gaussian distribution, we get:

\[ p(y_1 \ldots y_T | f^W(x)) = \prod_{i=1}^{T} p(y_i | f^W(x)) = \prod_{i=1}^{T} N(y_i; f^W(x), \sigma_i^2) \]

Note that \( \sigma \) is a model’s observation noise parameter.
Multi-task uncertainty-based likelihood

In maximum likelihood inference, we maximize the log likelihood of the model:

\[ L(W, \sigma, \ldots, \sigma) = -\log p(y_1, \ldots, y_T | f^W(x)) \]

Kendal et.al. (Kendal’2016) showed that:

\[ L(W, \sigma, \ldots, \sigma) = -\log p(y_1, \ldots, y_T | f^W(x)) \approx \sum_{i=1}^{T} \left( \frac{1}{2\sigma_i^2} L_i(W) + \log \sigma_i^2 \right) \]

- Proved that the formula can be extended for the binary cross entropy too (multi-label problems).

The MTL algorithm

1. Train the model
2. Estimate the loss $L$
3. Update the noise estimation $\sigma$
4. Check if the loss converged.
   - Yes
   - No

Validation
Our Food Dataset

- Food – 550 dishes, 11 categories, 11 cuisines
- Ingredients – 65
- Drinks – 40

In total:
more than
550,000 images

MAFood Data

Distribution of Categories in MAFood-110

Distribution of Cuisines in MAFood-110

Dataset available at: www.ub.edu/cvub/dataset

## Results

<table>
<thead>
<tr>
<th>Dish</th>
<th>GT</th>
<th>RUMTL</th>
<th>Single-task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tacos</td>
<td>Tacos</td>
<td>Prime Rib</td>
<td></td>
</tr>
<tr>
<td>Mexican</td>
<td>Mexican</td>
<td>American</td>
<td></td>
</tr>
<tr>
<td>Vegetable, meat, bread</td>
<td>Vegetable, bread</td>
<td>Vegetable, meat</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dish</th>
<th>GT</th>
<th>RUMTL</th>
<th>Single-task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eggs Benedict</td>
<td>Eggs Benedict</td>
<td>Ravioli</td>
<td></td>
</tr>
<tr>
<td>American</td>
<td>American</td>
<td>Italian</td>
<td></td>
</tr>
<tr>
<td>Vegetable, bread, egg</td>
<td>Vegetable, bread, egg</td>
<td>Vegetable, egg</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dish</th>
<th>GT</th>
<th>RUMTL</th>
<th>Single-task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sushi</td>
<td>Sushi</td>
<td>Cha Ca</td>
<td></td>
</tr>
<tr>
<td>Japanese</td>
<td>Japanese</td>
<td>Japanese</td>
<td></td>
</tr>
<tr>
<td>Vegetable, seafood, rice</td>
<td>Seafood, rice</td>
<td>Fried Food</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dish</th>
<th>GT</th>
<th>RUMTL</th>
<th>Single-task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ravioli</td>
<td>Bruschetta</td>
<td>Lobster Roll Sandwich</td>
<td></td>
</tr>
<tr>
<td>Italian</td>
<td>Italian</td>
<td>Italian</td>
<td></td>
</tr>
<tr>
<td>Dumpling</td>
<td>Vegetable, bread</td>
<td>Vegetable, meat, bread</td>
<td></td>
</tr>
</tbody>
</table>

Food ingredients recognition

Food category and class recognition
Neurons’ Activations

Ingredient activation: butter

Dish: creme brulee

Dish: creme brulee

Dish: creme brulee

Dish: scallops

Dish: creme brulee

Dish: cannoli

Dish: cannoli

Dish: french onion soup

Ingredient activation: granulated sugar

Dish: bibimbap

Dish: creme brulee

Dish: creme brulee

Dish: macarons

Dish: creme brulee

Dish: macarons

Dish: macarons

Dish: tacos

Dish: creme brulee

Ingredient activation: mayonnaise

Dish: hamburger

Dish: hamburger

Dish: hamburger

Dish: beet salad

Dish: hot dog

Dish: carrot cake

Dish: chicken curry

Dish: hamburger

Dish: carrot cake
Food ingredients recognition

Multi-task Accuracy: encourage errors to concentrate on the same data.
Contents

• The food image problem

• Multi-task food learning with aleatoric uncertainty

• Food recognition with epistemic uncertainty
  • - GAN
  • - Hierarchical classifier with epistemic uncertainty

• Conclusions
Bayesian neural networks

Instead of learning the model’s weights, learn a distribution over the weights

• => estimate uncertainty over the weights.

• So how do we do that?

Financial forecasting with probabilistic programming and Pyro
Bayesian Neural Networks

At inference, instead of taking the single set of weights that maximized the posterior distribution, we consider all possible weights, weighted by their probability.

\[
p(y|x, X, Y) = \int p(y|x, w)p(w|X, Y)dw
\]

- \(p(y|x, w)\) is the likelihood,
- \(p(w|X, Y)\) is the posterior probability of the model’s weights given the data.
Bayesian Neural Networks

But, how to compute the posterior probability of the model’s weights, $p(w|X,Y)$?

$$p(y|x, X, Y) = \int p(y|x, w)p(w|X, Y)dw$$
How to estimate the Epistemic Uncertainty?

Gal and Ghahramani showed that dropout at inference time gives an uncertainty estimator:

1. Infer $y|x$ multiple times, each time sample a different set of nodes to drop out.
2. Average the predictions to get the final prediction $E(y|x)$.
3. Calculate the sample variance of the predictions.
How to estimate the Epistemic Uncertainty?

The Epistemic Uncertainty (EU) can be expressed as follows:

where

\[
EU(x_t) = - \sum_{c=1}^{C} p(y_c = \hat{y}_c | x_t) \ln(p(y_c = \hat{y}_c | x_t)),
\]

K Monte Carlo dropout simulations

\[
p(y_c = \hat{y}_c | x) = \frac{1}{K} \sum_{k=1}^{K} p(y_c^k = \hat{y}_c^k | x).
\]
Class uncertainty

What to do with the difficult classes?

Are all classes well represented/easily discriminable?

How to augment difficult classes?

- data augmentation

Adapted from Gal (2016)
Use Uncertainty for Data Augmentation

<table>
<thead>
<tr>
<th>Brightness</th>
<th>Color</th>
<th>Contrast</th>
<th>Sharpness</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Brightness augmentation
- Color augmentation
- Contrast augmentation
- Sharpness augmentation
- All augmentations combined
Sample of the synthetic images from the generator applied.
Class uncertainty

What to do with the difficult classes?

Are all classes well represented/easily discriminable?

How to augment difficult classes?
- classic data augmentation, or
- creating synthetic images. How?

Adapted from Gal (2016)
The Biggest Breakthrough In The History Of AI

- Celebrated computer scientist Yann Lecun observed:

  “GANs and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.”

VP and Chief AI Scientist, Facebook
Silver Professor of Computer Science, Data Science, Neural Science, and Electrical and Computer Engineering, New York University.
ACM Turing Award Laureate, (sounds like I'm bragging, but a condition of accepting the award is to write this next to your name)
Member, National Academy of Engineering
Generative Adversarial Network (GAN)

New components:
- Transposed convolution,
- Batch Normalization

Binary Classifier:
- Conv, Leaky ReLU,
- FC, Sigmoid

https://github.com/PramodShenoy/GANerations
GAN

Loss function for D
If $x$ is real, $D(x) = 1$; otherwise, $D(x) = 0$
Minimize the error

$$L_D = \mathbb{E}_x \ln D(x) + \mathbb{E}_z \ln (1 - D(G(z)))$$

Real
$D(x) \to 1$
$\Rightarrow \ln D(x) \to 0$

Fake
$D(x) \to 0$
$\Rightarrow \ln (1 - D(x)) \to 0$

Loss function for G
Maximize the error of D

Minimax procedure

$$\min_D \max_G \mathbb{E}_x \ln D(x) + \mathbb{E}_z \ln (1 - D(G(z)))$$
Use the data augmentation applied class-conditionally to improve the results in terms of accuracy and also to reduce the overall epistemic uncertainty.

During the prediction phase, the same image is fed to the CNN several times to calculate the epistemic uncertainty given by the model for that image.

Validation
SinGAN’s multi-scale pipeline: the model consists of a pyramid of GANs, where both training and inference are done in a coarse-to-fine fashion. At each scale, $G_n$ learns to generate image samples in which all the overlapping patches cannot be distinguished from the patches in the down-sampled training image, $x_n$, by the discriminator $D_n$; the effective patch size decreases as one goes up the pyramid (marked in yellow on the original image for illustration). The input to $G_n$ is a random noise image $z_n$, and the generated image from the previous scale $\sim x_n$, upsampled to the current resolution (except for the coarsest level which is purely generative).

Use Uncertainty for Data Augmentation

Original Images

Synthetic Images

Synthetic image generated on the selected images from the training set
AttentionGAN: Unpaired Image-to-Image Translation using Attention-Guided Generative Adversarial Networks

Hao Tang, Hong Liu, Dan Xu, Philip H.S. Torr and Nicu Sebe

Framework of the proposed attention-guided generation scheme I, which contains two attention-guided generators G and F. One mapping is shown: x→G(x)→F(G(x))→x. The other mapping is: y→F(y)→G(F(y))→y. The attention-guided generators have a built-in attention module, which can perceive the most discriminative content between the source and target domains. The input image, the content mask and the attention mask are fused to synthesize the final result.
Using AttentionGan

Use Uncertainty for Data Augmentation

Number of synthetic images generated after the third training cycle.

Histogram for the entropy of the predicted images
Use Uncertainty for Data Augmentation

Training images vs epistemic uncertainty
Use Uncertainty for Data Augmentation

Results on Food-101 in terms of Acc and NEU for the models trained with different data augmentation techniques.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>NEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>61.00%</td>
<td>30.22%</td>
</tr>
<tr>
<td>ResNet50+DA</td>
<td>65.02%</td>
<td>33.55%</td>
</tr>
<tr>
<td>ResNet50+DA+A</td>
<td>64.65%</td>
<td>36.53%</td>
</tr>
<tr>
<td>Proposed method</td>
<td>65.54%</td>
<td>33.51%</td>
</tr>
</tbody>
</table>

Results on UECFOOD-256 in terms of Acc and NEU for the models trained with different data augmentation techniques.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>NEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>77.66%</td>
<td>19.85%</td>
</tr>
<tr>
<td>ResNet50+DA</td>
<td>82.65%</td>
<td>27.35%</td>
</tr>
<tr>
<td>ResNet50+DA+A</td>
<td>82.54%</td>
<td>29.45%</td>
</tr>
<tr>
<td>Proposed method</td>
<td>82.82%</td>
<td>26.25%</td>
</tr>
</tbody>
</table>
# Use Uncertainty for Data Augmentation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ResNet50 ($S_1$)</th>
<th>ResNet50 ($S_2$)</th>
<th>ResNet50 ($S_3$)</th>
<th>ResNet50 ($S_4$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>81.99%</td>
<td>83.69%</td>
<td>84.10%</td>
<td>84.26%</td>
</tr>
<tr>
<td>Chinese</td>
<td>87.93%</td>
<td>90.05%</td>
<td>90.60%</td>
<td>91.17%</td>
</tr>
<tr>
<td>French</td>
<td>89.01%</td>
<td>90.33%</td>
<td>94.12%</td>
<td>92.54%</td>
</tr>
<tr>
<td>Greek</td>
<td>89.12%</td>
<td>89.34%</td>
<td>89.90%</td>
<td>92.11%</td>
</tr>
<tr>
<td>Indian</td>
<td>87.67%</td>
<td>92.96%</td>
<td>93.29%</td>
<td>92.41%</td>
</tr>
<tr>
<td>Italian</td>
<td>80.72%</td>
<td>82.44%</td>
<td>84.31%</td>
<td>84.07%</td>
</tr>
<tr>
<td>Japanese</td>
<td>88.08%</td>
<td>90.85%</td>
<td>91.20%</td>
<td>90.93%</td>
</tr>
<tr>
<td>Mexican</td>
<td>79.12%</td>
<td>80.37%</td>
<td>81.64%</td>
<td>81.96%</td>
</tr>
<tr>
<td>Thai</td>
<td>70.98%</td>
<td>79.91%</td>
<td>79.85%</td>
<td>79.22%</td>
</tr>
<tr>
<td>Turkish</td>
<td>91.44%</td>
<td>91.65%</td>
<td>91.92%</td>
<td>92.15%</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>84.67%</td>
<td>86.99%</td>
<td>88.14%</td>
<td>89.85%</td>
</tr>
</tbody>
</table>

Results obtained on the test sets in terms of $R_{macro}$.

\[
R_{macro}(\hat{y}, \bar{y}) = \frac{1}{|C|} \sum_{c \in C} R_{micro}(y_c, \bar{y}_c)
\]
How many dishes are there all over the world?

More than 100,000 basic foods
Imagine

- When you visit Mexico,
  what is the probability to eat a food from Norway?
Let’s organize classes in meta-classes
Let’s organize classes in meta-classes
But .... Hierarchical classifiers have a big problem

Hypothesis: use uncertainty to decide if a LPN should be used
Proposed Method

Validation
Dataset complementary to Food101:
• 101 classes / dishes
• 1000 images per class

A recipe for each downloaded resulting in a list of ingredients per class and a total of 446 unique ingredients.

A total of 279 different ingredients were considered, visible or not, with an average of 19 per dish.
Ablation study

- Hierarchical w/o constraints: 79.96
- Flat: 81.37
- Hierarchical w/ constraints: 81.62

- pred_wd + pred_wod: 81.11
- pred_wd + eu: 81.55
- pred_wd + eu: 81.27
- pred_wd + pred_wod + eu: 81.62
Results - Samples of the Smallest and Largest EU within the same class of Dish

- Caesar Salad
- Ravioli
- Steak
- Tacos
Conclusions

• Food image world brings us huge amount of data and Computer Vision questions

• Transfer learning and its subproblems (multi-task learning) open new opportunities

• Uncertainty modeling is a hot topic with many open questions and challenges!
  • Exclusivity relation between elements helps to the classification

• Epistemic uncertainty
  New method for robust hierarchical classifiers.
  A good cue to improve recognition scalability.
  Epistemic uncertainty useful beyond the confidence of the model.

• Aleatoric uncertainty
  Allows to weight different tasks according to uncertainty

• For first time a food ontology is integrated into an end-to-end model

A huge impact of food analysis is expected from point of view of:
• Science, but also
• Real world applications, specially important for the society.
Thank you!