Object Recognition as Multimedia Translation

Peter Carbonetto
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada

Nando de Freitas
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada

{pcarbo,nando}@cs.ubc.ca

Kobus Barnard
Computer Science
387 Soda Hall, Berkeley
CA 94720-1776, USA

Pinar Duygulu
Computer Science
387 Soda Hall, Berkeley
CA 94720-1776, USA

David Forsyth
Computer Science
387 Soda Hall, Berkeley
CA 94720-1776, USA

{kobus,pinar,daf}@cs.berkeley.edu

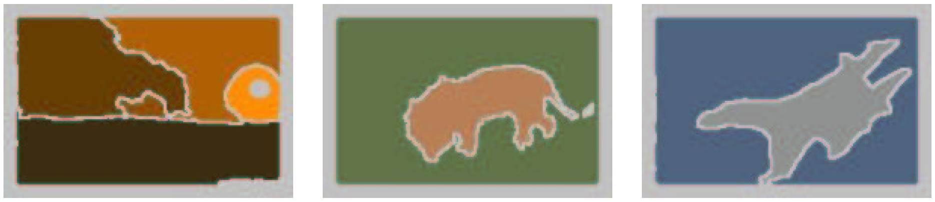
Abstract

We present an approach for object recognition as statistical multimedia translation. In this approach, recognition is a process of annotating image regions with words. For training data, we use large databases containing documents with images and associated text. Images are segmented into regions (blobs) using several image features. We introduce a probabilistic model that provides a mapping between continuous blob vectors and keywords supplied with the images. The model is learned by maximising the log-likelihood of the data. This process is analogous with learning a lexicon from an aligned bitext, except that it allows for the incorporation of continuous random variables.

1 Introduction

There is a wide variety of data-sets that consist of very large numbers of annotated images. Examples include the Corel data-set (see Figure 1), most museum image collections (e.g. `http://www.thinker.org/fam/thinker.html`), the web archive (`http://www.archive.org`), and most collections of news photographs on the web (which come with captions). Typically, these annotations refer to the content of the annotated image, more or less specifically and more or less comprehensively. For example, the Corel annotations describe specific image content, but not all of it; museum collections are often annotated with some specific material — the artist, date of acquisition, etc. — but often contain some rather abstract material as well.

This data allows us to formulate object recognition as a statistical machine translation problem [1]. In particular, we segment images into regions (blobs) and then learn to predict words using regions. Each region is described by some set of features. In machine translation, a lexicon links discrete objects (words in one language) to discrete objects (words in the other language) [2, 3]. However, the features naturally associated with image regions do not occupy a discrete space. The simplest solution to this problem is to use K-means to



sea sky sun waves

cat forest grass tiger

jet plane sky

Figure 1: *Examples from the Corel data set. We have associated keywords and segments for each image, but we don't know which word corresponds to which segment. The number of words and segments can be different; even when they are same, we may have more than one segment for a single word, or more than one word for a single blob. We try to align the words and segments, so that for example an orange stripy blob will correspond to the word tiger.*

vector quantise the image region representation. This approach, adopted in [1], was well received by the computer vision community (best paper prize in cognitive computer vision at ECCV 2002). However, we would like to surmount the problem of having to introduce noise by vector quantising the blobs. Consequently, we propose a discrete-Gaussian translation model that allows us to translate Gaussian features directly to discrete features.

2 Translation models

Our approach to recognition is analogous to machine translation. We have a representation of one form (image regions; French) and wish to turn it into another form (words; English). In particular, our models act as *lexicons*, devices that predict one representation (words; English) given another representation (image regions; French). Learning a lexicon from data is a standard problem in machine translation literature [2, 3]. Typically, lexicons are learned from a form of data-set known as an **aligned bitext** — a text in two languages where rough correspondence, perhaps at the paragraph or sentence level, is known. The problem of lexicon acquisition involves determining precise correspondences between words of different languages. Data-sets consisting of annotated images are aligned bitexts — we have an image consisting of regions, and a set of text. While we know the text goes with the image, we don't know which word goes with which region.

We have a "chicken and egg" situation. To build the translation probabilities, we need to know the correct associations (correspondences). To estimate the correspondences, we need to know the translation probabilities. As explained in [1], this problem can be surmounted by marginalising over the possible associations using a mixture model proposed for text in [2]. This model assumes one-to-one assignments. Using the notation of Figure 2, the mathematical expression for the model is:

$$p(\mathbf{b}|\mathbf{w}) = \prod_{n=1}^{N} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) t(b = b_{nj} | w = w_{ni}). \tag{1}$$

This mixture is illustrated in Figure 3. Note that since the blob order is random, there is no reason for preferring specific alignments. We, therefore, set $p(a_{nj} = i) = 1/L_n$.

In [1], $t(b = b_{nj} | w = w_{ni})$ is a discrete translation table. In our case, the translation probabilities $t(\cdot|\cdot)$ are Gaussian instead of discrete. In particular, there is one Gaussian for each word in the vocabulary. This model is fairly intuitive. Each word generates image segments from a corresponding Gaussian cluster. By clustering the words (say with K-means or SVD methods), it is also possible to have a Gaussian cluster for each concept.

N	Number of images.		
$\mathbf{w}$	Words in the databases.	$\mathbf{b}$	Blobs in the database.
w_n	Words in the n-th image.	b_n	Blobs in the n-th image.
	$w_n = (w_{n1}, \ldots, w_{ni}, \ldots, w_{nL_n})$		$b_n = (b_{n1}, \ldots, b_{nj}, \ldots, b_{nM_n})$
$w^\star$	A particular word.	$b^\star$	A particular blob.
L_T	Number of words.	M_T	Number of blobs.
L_n	Number of words in the n-th image.	M_n	Number of blobs in the n-th image.

a_n	Assignment $a_n = \{a_{n1}, \ldots, a_{nM_n}\}$, $a_{nj} = i$ if b_{nj} translates to w_{ni}.
$t(b\|w)$	Translation distribution.
$p(a_{nj} = i)$	Assignment probabilities.
θ	Set of model parameters.
$p(a_{nj} = i\|w_{ni}, b_{nj}, \theta^{(\mathrm{old})})$	Indicator probabilities.

Figure 2: *Notation.*

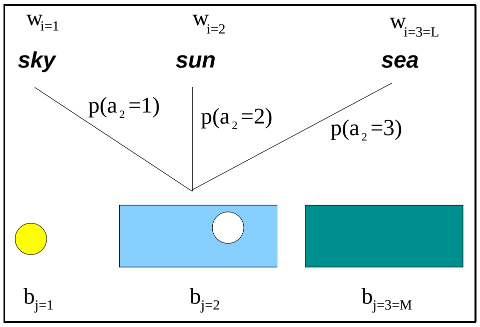


Figure 3: *Example: Each blob is predicted with some probability by each word, meaning that we have a mixture model over the words. The association probabilities provide the correspondences (assignments) between each word and the various image segments (blobs).*

The discrete-Gaussian translation model is:

$$p(\mathbf{b}|\mathbf{w}) = \prod_{n=1}^{N} L_n^{-M_n} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} t(b_{nj}|w_{ni}) = \prod_{n=1}^{N} L_n^{-M_n} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} \mathcal{N}(b_{nj}; \mu_k, \Sigma_k)\delta_{w_k^\star}(w_{ni}) \tag{2}$$

where $\delta_{w_k^\star}(w_{ni}) = 1$ if $w_k^\star$ $(k = 1, \ldots, L_T)$ appears in the n-th annotation and is 0 otherwise. Note that we can use Bayes rule to reverse the order of translation, $p(\mathbf{w}|\mathbf{b}) \propto p(\mathbf{b}|\mathbf{w})p(\mathbf{w})$. In the following section, we derive maximum likelihood (ML) algorithms to learn the parameters $\theta = \{\mu, \Sigma\}$ of this mixture model of known mixing proportions.

3 Maximum Likelihood Estimation with EM

We want to find the maximum likelihood parameters

$$\theta^{\mathrm{ML}} = \underset{\arg\max}{\theta} \; p(\mathbf{w}|\mathbf{b},\theta) = \underset{\arg\max}{\theta} \; \sum_{a} p(\mathbf{a},\mathbf{w}|\mathbf{b},\theta). \tag{3}$$

We can carry out this optimisation using an EM algorithm [4], which iterates between the following two steps:

1. E step: Compute the expectation of the complete log-likelihood with respect to the distribution of the assignment variables $Q^{\mathrm{ML}} = \mathbb{E}_{p(\mathbf{a}|\mathbf{w},\mathbf{b},\theta^{(\mathrm{old})})}\left[\log p(\mathbf{a},\mathbf{b}|\mathbf{w},\theta)\right]$, where $\theta^{(\mathrm{old})}$ refers to the value of the parameters at the previous time step.

2. M step: Find the new maximum $\theta^{(\mathrm{new})} = \underset{\arg\max}{\theta} Q^{\mathrm{ML}}$.

In our case, the Q^{ML} function is given by

$$Q^{\mathrm{ML}} = \sum_{n=1}^{N}\sum_{j=1}^{M_n}\sum_{i=1}^{L_n} p(a_{nj}=i|b_{nj},w_{ni},\theta^{(\mathrm{old})}) \log\left[\frac{1}{L_n} t(b_{nj}|w_{ni})\right].$$

where the posterior associations, $p(a_{nj}=i|b_{nj},w_{ni},\theta^{(\mathrm{old})})$, are given by

$$p(a_{nj}=i|b_{nj},w_{ni},\theta^{(\mathrm{old})}) = \frac{\frac{1}{L_n} t(b_{nj}|w_{ni})}{\sum_{i'=1}^{L_n} \frac{1}{L_n} t(b_{nj}|w_{ni'})} = \frac{t(b_{nj}|w_{ni})}{\sum_{i'=1}^{L_n} t(b_{nj}|w_{ni'})}$$

Maximisation of the Q^{ML} function is now straightforward, and leads to the EM algorithms shown in Figures 4 and 5.

Initialise (Cluster all image regions with K-means)

E step For each $n = 1,\ldots,N$, $j = 1,\ldots,M_n$ and $i = 1,\ldots,L_n$, compute

$$p(a_{nj}=i|w_{ni},b_{nj},\theta^{(\mathrm{old})}) = \frac{t(b_{nj}|w_{ni})}{\sum_{i'=1}^{L_n} t(b_{nj}|w_{ni'})}$$

M step For each pair $(b^\star, w^\star)$ appearing together in at least one of the images, compute

$$\widetilde{t}(w_{ni}=w^\star|b_{nj}=b^\star) = \sum_{n=1}^{N}\sum_{j=1}^{M_n}\sum_{i=1}^{L_n} p(a_{nj}=i|w_{ni},b_{nj},\theta^{(\mathrm{old})})\delta_{(w^\star,b^\star)}(w_{ni},b_{nj})$$

where $\delta_{(w^\star,b^\star)}(w_{ni},b_{nj})$ is 1 if $b^\star$ and $w^\star$ appear in image and 0 otherwise.

Normalise $\widetilde{t}(w_{ni}=w^\star|b_{nj}=b^\star)$ to obtain $t(w_{ni}=w^\star|b_{nj}=b^\star)$.

Figure 4: *EM algorithm for discrete-discrete translation.*

When implementing the discrete-Gaussian algorithm, it is a good idea to rewrite the mean and covariance update equations as

$$\begin{aligned}\mu_k &= T_1^{-1}T_2\\ \Sigma_k &= T_1^{-1}\left(T_3 - T_1^{-1}T_2T_2'\right)\end{aligned}$$

Initialise

E step For each $n = 1, \dots, N$, $j = 1, \dots, M_n$ and $i = 1, \dots, L_n$, compute

$$p(a_{nj} = i | w_{ni}, b_{nj}, \theta^{(\text{old})}) = \frac{t(b_{nj} | w_{ni})}{\sum_{i'=1}^{L_n} t(b_{nj} | w_{ni'})}$$

M step Update the means and covariances

$$\mu_k = \frac{\sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i | b_{nj}, w_{ni}, \theta^{(\text{old})}) \;\; b_{nj} \;\; \delta_{w_k^\star}(w_{ni})}{\sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i | b_{nj}, w_{ni}, \theta^{(\text{old})}) \delta_{w_k^\star}(w_{ni})}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i | b_{nj}, w_{ni}, \theta^{(\text{old})}) (b_{nj} - \mu_k)(b_{nj} - \mu_k)' \delta_{w_k^\star}(w_{ni})}{\sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i | b_{nj}, w_{ni}, \theta^{(\text{old})}) \delta_{w_k^\star}(w_{ni})}$$

where $\delta_{w_k^\star}(w_{si})$ emphasizes that one carries out the sum for the different tokens $w_k^\star$ in the vocabulary.

Figure 5: *EM algorithm for discrete-Gaussian translation.*

where

$$\begin{aligned} T_1 &\triangleq \sum_{s=1}^{S} \sum_{j=1}^{M_s} \sum_{i=1}^{L_s} p(a_{sj} = i | b_{sj}, w_{si}, \theta^{(\text{old})}) \delta_{w_k^\star}(w_{si}) \\ T_2 &\triangleq \sum_{s=1}^{S} \sum_{j=1}^{M_s} \sum_{i=1}^{L_s} p(a_{sj} = i | b_{sj}, w_{si}, \theta^{(\text{old})}) \;\; b_{sj} \;\; \delta_{w_k^\star}(w_{si}) \\ T_3 &\triangleq \sum_{s=1}^{S} \sum_{j=1}^{M_s} \sum_{i=1}^{L_s} p(a_{sj} = i | b_{sj}, w_{si}, \theta^{(\text{old})}) \;\; b_{sj} b'_{sj} \;\; \delta_{w_k^\star}(w_{si}) \end{aligned}$$

4 Experiments

4.1 Evaluation

The design of an objective performance evaluation is not an easy task. Our goal is to answer the question — do we predict appropriate words for each particular region? One could try to answer this question by looking at the pictures. This form of manual evaluation is not only subjective, but also very difficult to carry out for large data-sets. A less strict, but nonetheless informative test, is to determine how well we can annotate images. That is, to check whether the models predict the words appearing in the Corel annotations. Corel annotations typically omit some obviously appropriate words. However, since our purpose is to compare methods this is not a significant problem as each model must cope with the same set of missing annotations. Performance comparisons can then be carried out automatically and therefore on a substantial scale.

We also compare the models' predictions with respect to the predictions obtained using the empirical word frequency of the training set. Matching the performance of the empirical density is required to demonstrate non-trivial learning. Doing substantially better than this on the Corel data is difficult. The annotators typically provide several common words (*e.g. sky, water, people*), and fewer less common words (*e.g. tiger*). This means that annotating all images with, say, *sky*, *water*, and *people* is quite a successful strategy. Performance using the empirical word frequency would be reduced if the empirical density was flatter. Thus, the increment of performance over the empirical density is a sensible indicator. We look at word prediction on held-out data (165 images), and rank models using three measures:

PR-1: This measure scores the number of times the predicted word of highest probability corresponds to one of the Corel annotations. The range of this score is clearly from 0 to 1.

PR-L: This measure scores the number of times the L_n predicted words of highest probability match the L_n annotations. Thus if there are three keywords in the n-th image (*e.g. sky*, *water*, and *sun*), then $L_n = 3$, and we compare these words to the three predicted words of highest probability.

NS: On the whole, a well-fit model should predict a better set of words, but we need some concrete measurement of the appropriateness of the omitted words. The difficulty here is that one needs a loss function, and traditional zero-one loss is highly misleading. For most conceivable applications, certain errors (*cat* for *tiger*) are less offensive than others (*car* for *vegetable*). Because the number of classes that we can predict is large (the size of the vocabulary), we normalize the correct and incorrect classifications. Specifically, we compute $NS = c/L_n - i/(L_T - L_n)$, where L_T is the vocabulary size, L_n is the number of actual words for the image, c is the number of words predicted correctly, and i is the number of words predicted incorrectly. This score gives a value of 0 for both predicting everything and predicting nothing, and 1 for predicting exactly the actual word set (no false positives, no false negatives). The score for exactly predicting the complement of the actual word set is -1. A more comprehensive study of evaluation measures for this problem is presented in [5].

4.2 Results

We trained the models on 420 annotated images from the Corel database. Each image has 4-5 keywords. Images are segmented using Normalized Cuts [6]. For discrete-discrete translation regions are clustered into 63 blobs (the number of words in the vocabulary) using K-means. We use 25 features for each region (including region color and standard deviation, region average orientation energy (12 filters), region size, and position). We choose a consistent set of features, as we wish to study mechanisms of recognition rather than specific feature sets. Table 1 shows the scores obtained in this experiment. It is easy to see that both translations models outperform the empirical distribution consistently, yet it is hard to say which of the two models performs better.

Table 1: *Annotation prediction scores for a simple experiment. A larger number indicates better performance.*

METHOD	DATA-SET	PR-1	PR-L	NS
Empirical	train	0.179	0.150	0.446
Discrete-discrete	train	0.321	0.354	0.549
Discrete-Gaussian	train	0.538	0.398	0.554
Empirical	test	0.145	0.142	0.436
Discrete-discrete	test	0.313	0.322	0.509
Discrete-Gaussian	test	0.470	0.327	0.470

Figures 6 and 7 show chosen results on the train and test data sets. Figure 8 shows results on three images that were sampled randomly from the test set.

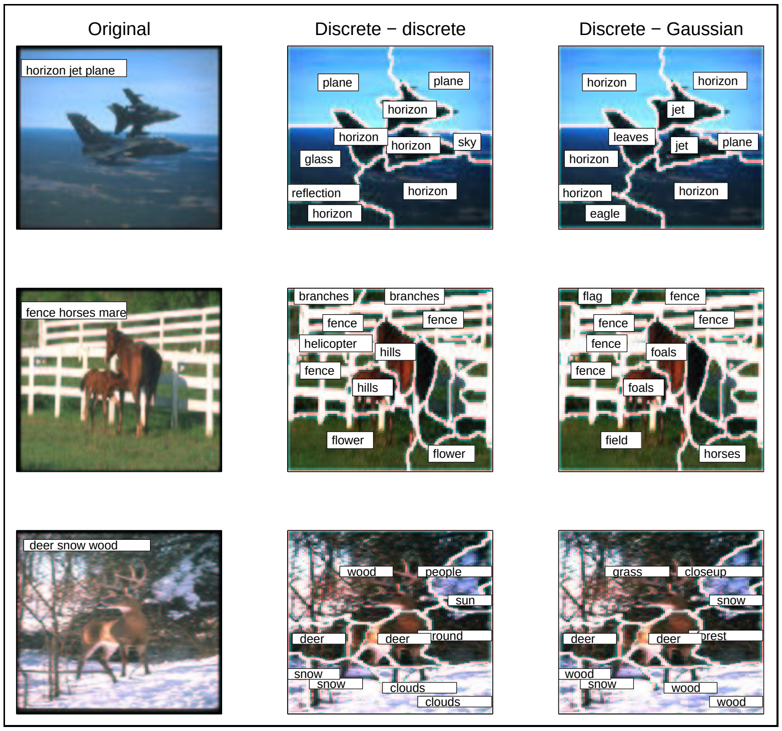


Figure 6: *Chosen recognition results on the train set.*

5 Conclusions

We introduced a model for discrete-Gaussian multimedia translation. Early results indicate similar performance to the discrete models, based on quantisation, on a difficult recognition problem. However, the single-stage discrete-Gaussian model is not only a more elegant solution than the two-stage discrete-discrete model, it also opens up room for new improvements. In particular, it enables us to implement feature selection algorithms (not reported in this paper). In addition, simplifying the structure of the covariances can result in sparser models and more computationally efficient algorithms. The results should also improve by adopting appropriate Gauss-Wishart priors. Although we focused on Gaussians, this approach applies to other continuous distributions from the exponential family.

References

[1] P Duygulu, K Barnard, N de Freitas, and D Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[2] P Brown, S A Della Pietra, V J Della Pietra, and R L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[3] I D Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001.

[4] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

[5] K Barnard, P Duygulu, N de Freitas, D Forsyth, D Blei, and M I Jordan. Matching words and pictures. *Journal of Machine Learning Research.* Under review, 2002.

[6] J Shi and J Malik. Normalized cuts and image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

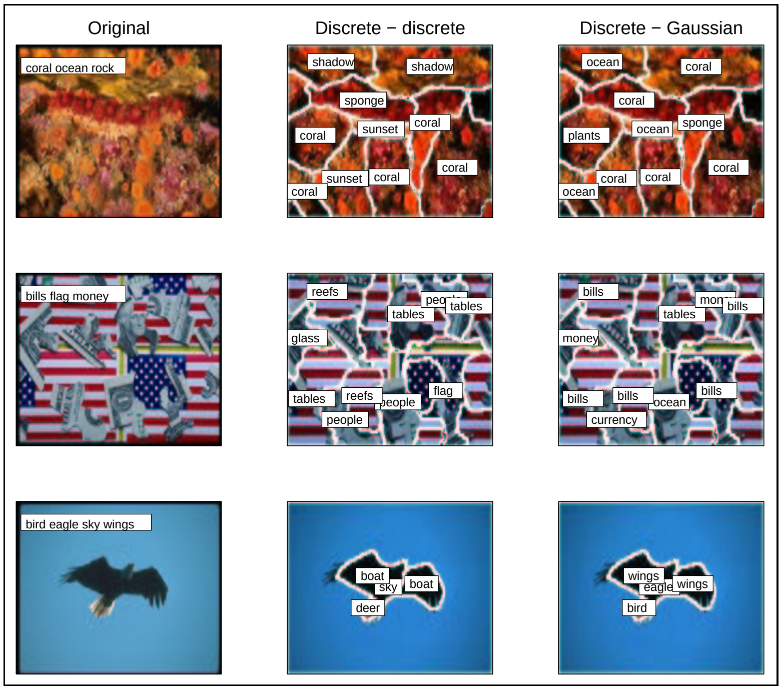


Figure 7: *Chosen recognition results on the test set.*

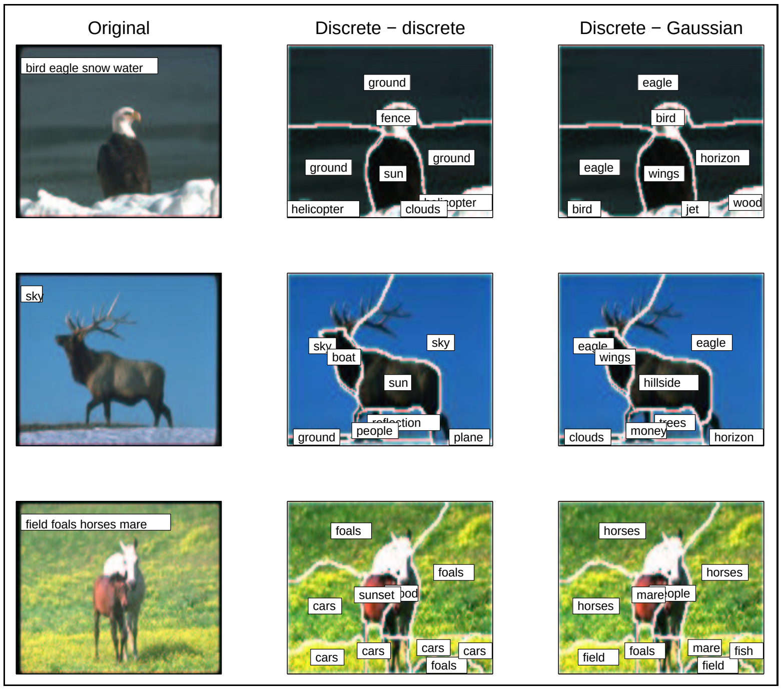


Figure 8: *Randomly sampled results on the test set.*