Towards Transparent Systems: Semantic Characterization of Failure Modes

Aayush Bansal[1], Ali Farhadi[2], and Devi Parikh[3]

[1]Carnegie Mellon University [2]University of Washington [3]Virginia Tech

1 Introduction

Today's computer vision systems are not perfect. They fail frequently. Even worse, they fail abruptly and seemingly inexplicably. We argue that making our systems more transparent via an explicit human understandable characterization of their failure modes is desirable. We propose characterizing the failure modes of a vision system using semantic attributes. For example, a face recognition system may say "If the test image is blurry, or the face is not frontal, or the person to be recognized is a young white woman with heavy make up, I am likely to fail." This information can be used at training time by researchers to design better features, models, or collect more focused training data. It can also be used by a downstream machine or human user at test time to know when to ignore the output of the system, in turn making it more reliable. To generate such a "specification sheet", we discriminatively cluster incorrectly classified images in the semantic attribute space using L1-regularized weighted logistic regression. We show that our specification sheets can predict oncoming failures for face and animal species recognition better than several strong baselines. We also show that lay people can easily follow our specification sheets.

2 Approach

While our approach can be applied to any vision system, we use image classification as a case study in this paper. We are given a set of N images along with their corresponding class labels $\{(\boldsymbol{x}_i, y'_i)\}, i \in \{1, \ldots, N\}, y' \in \{1, \ldots, C\}$, where C is the number of classes. We are also given a pre-trained classification system $H(\boldsymbol{x})$ whose failures we wish to characterize. Given an image $\boldsymbol{x}_i$, the system predicts a class label $\hat{y}'_i$ for the image i.e. $\hat{y}'_i = H(\boldsymbol{x}_i)$. We assign each image in our training set to a binary label $\{(\boldsymbol{x}_i, y_i)\}, y_i \in \{0, 1\}$, where $y_i = 0$ if $\hat{y}'_i = y'$ i.e. images $\boldsymbol{x}_i$ is correctly classified by H, otherwise $y_i = 1$. We annotate all images $\boldsymbol{x}_i$ using a vocabulary of M binary attributes $\{a_m\}, m \in \{1, \ldots, M\}$. Each image is thus represented with an M dimensional binary vector i.e. $\boldsymbol{x}_i \in \{-1, 1\}^M$ indicating whether attribute a_m is present in the image or not. We wish to discover a specification sheet, which we represent as a set of sparse lists of attributes – each list capturing a cluster of incorrectly classified or "mistake" images i.e. a failure mode.

Discriminative Clustering: We discriminatively cluster the mistake images in this ground truth attributes space. We initialize our clustering using k-means. This gives each of the mistake images a cluster index $c_i \in \{1, \ldots, K\}$. We denote

all mistake images belonging to cluster k as $\{\boldsymbol{x}_i^k\}$. We train a discriminative function $h_k(\boldsymbol{x}_i)$ for each of the clusters that separates $\{\boldsymbol{x}_i^k\}$ from other "negative" images.

L1-Regularized Logistic Regression: The discriminative function we train for each cluster is an L1-regularized logistic regression. It is trained to separate mistake images belonging to cluster k ($y_i^k = 1$) from all not-mistake images ($y_i^k = 0$). y_i^k is the label assigned to images for training the cluster-specific discriminative function. Notice that here y_i^k is not defined for images belonging to other mistake clusters $\boldsymbol{x}_i^l, l \in \{1, \ldots, K\}, l \neq k$, as they do not participate in training the discriminative function for cluster k. All discriminative functions share the same negative set i.e. the not-mistake images $\{\boldsymbol{x}_i^0\}$.

Weighted Logistic Regression: In addition to identifying attributes that separate mistake from not-mistake images, we also wish to ensure that images belonging to the same cluster share many attributes in common and more importantly, the attributes selected to characterize the clusters are present in most of the images assigned to that cluster. This will help make the specification sheet accurate and precise. To encourage this, rather than using a standard L1-regularized logistic regression as described above, we use a weighted logistic regression. At each iteration, we replace each binary attribute in the image representation with the proportion of images in the cluster that share the same (binary) attribute value.

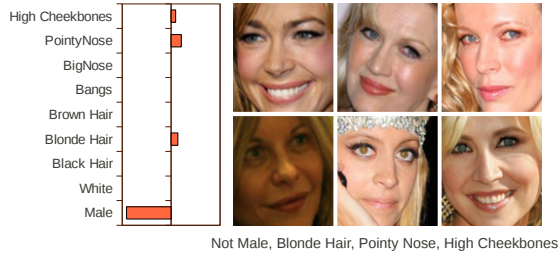

Fig. 1. The learnt sparse discriminative function for each cluster can be directly converted to a compact semantic description of the cluster.

Hierarchical Clustering: In addition to the weighted discriminative clustering described above, we also experiment with hierarchical clustering. Given a branching factor B, we initialize the clustering using k-means with B clusters. We run the iterative discriminative clustering approach described above till convergence using weighted L1-regularized logistic regression. We then further cluster each of the B clusters into B clusters using the same iterative discriminative clustering, and so on, till the tree reaches a predetermined depth D.

3 Experiments

Datasets: We experiment with two domains: face (celebrity) and animal species recognition. For faces, 2400 images from 60 categories (40 images per category) from the development set of the Public Figures Face Database (Pubfig) of Kumar *et al.* [1] are used. For animals, 1887 images from 37 categories (51 images per category) from the Animals with Attributes dataset (AwA) of Lampert *et al.* [2] containing 85 (annotated) attributes are used.

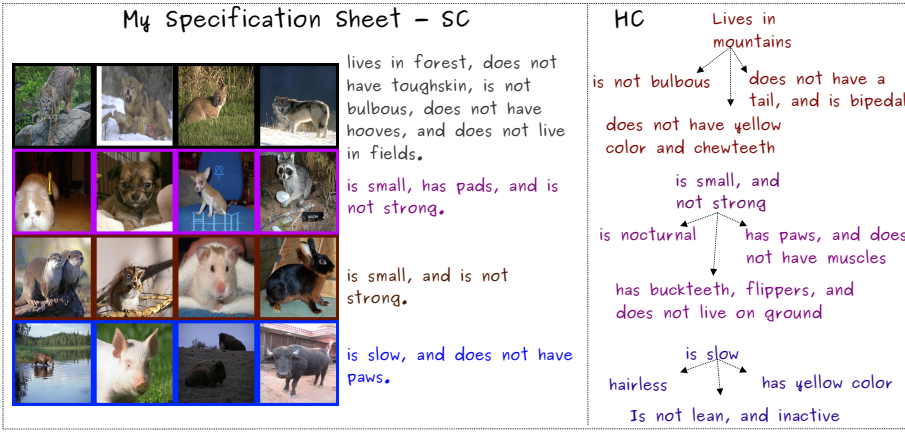

Fig. 2. Example specification sheets generated by our approach. Left: Simple clustering (SC): The failure modes are listed. For illustration, we show example images belonging to each cluster. Right: Hierarchical clustering (HC): Each path leading to a leaf is a failure mode *e.g.* "is slow and has yellow color" for the right most leaf of the bottom tree. Best viewed in color.

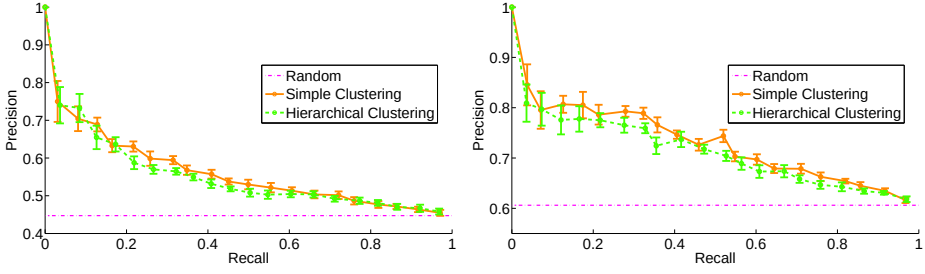

Fig. 3. Performance of our generated specification sheets capture failures. Left: Pubfig, Right: AwA

Failure Prediction: We can predict failure using ground truth attributes (mimicking a user using our specification sheets at test time), as well as automatically by combining failure predictions of multiple specification sheets using pre-trained attribute predictors. Note that automatic failure prediction can be thought of as a classifier confidence estimate.

Metric: We evaluate the ability of our specification sheets to predict failure using precision and recall (PR), where we evaluate how often an image predicted by the specification sheet to be a failure truly is a failure (precision), and what percentage of the true failures are detected by the specification sheet (recall).

Baselines: We compare our automatic failure prediction approach to other non-semantic baselines. **ClassConf (CC):** The conventional approach to estimating the confidence of a classifier is computing the entropy of the probabilistic output of the classifier across the class labels (*e.g.* computed using Platts' method [3]) to a given test instance. **Boost:** Our approach to automatic failure prediction

	Random	SC - all	SC - sel	HC - all	HC - sel
Pubfig	0.4473	0.5473	0.5421	0.5370	0.5291
AwA	0.6061	0.7088	0.7079	0.6942	0.6963

	CC	Boost	SC	HC	Rand
Pubfig	0.64	0.64	0.68	0.68	0.45
AwA	0.77	0.74	0.77	0.77	0.61

Table 1. Area under the precision recall (PR) curve for different approaches. CC: ClassConf, SC: simple clustering, HC: hierarchical clustering, all: using all attributes, sel: using a subset of attributes that are easy for lay people to understand. **Left:** Comparison of various approaches when using ground truth attributes at test time (mimicking human user in the loop). **Right:** Comparison of various approaches to automatic failure prediction.

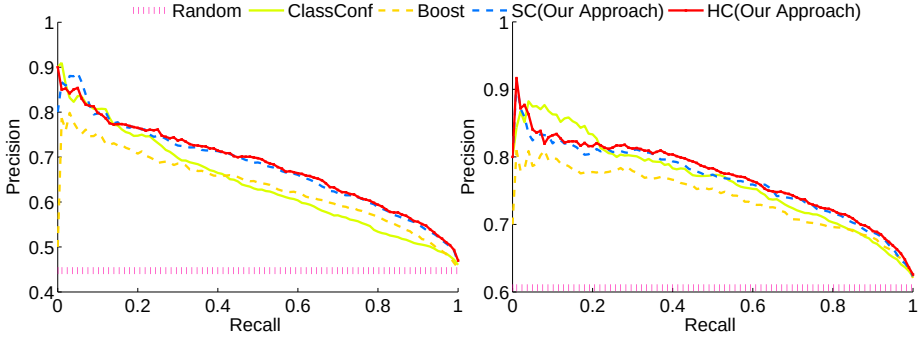

Fig. 4. Performance of our specification sheets at *automatically* predicting oncoming failure. Left: Pubfig, Right: AwA.

employs multiple classifiers. This is related to boosting approaches [4]. We use Adaboost [5] to learn the weights of 2000 decision trees. **Rand:** We also compare to a baseline that assigns each image a random score between [0,1] as a likelihood of failure. As seen in Table 1, and Figure 3 and Figure 4, our approach outperforms these baselines.

4 Conclusion

We proposed a discriminative clustering approach using L1-regularized weighted logistic regression to generate semantically understandable "specification sheets" that describe the failure modes of vision systems. We presented promising results for failure prediction in face and animal species recognition. We demonstrated that the specification sheets capture failure modes well, and can be leveraged to automatically predict oncoming failure better than a standard classifier confidence measure and a boosting baseline. By being better informed via our specification sheets, researchers can design better solutions to vision systems, and users can choose to not use the vision system in certain scenarios, increasing the performance of the system when it is used. Downstream applications can also benefit from our automatic failure prediction.

References

1. Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and simile classifiers for face verification. In: ICCV. (2009)

2. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR. (2009)
3. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Advances in Large Margin Classiers. (2000)
4. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Machine Learning International Workshop. (1996)
5. Appel, R., Fuchs, T., Dollár, P., Perona, P.: Quickly boosting decision trees - pruning underachieving features early. In: ICML. (2013)