Detecting Anomalous Faces with "No Peeking" Autoencoders

Anand Bhattad, Jason Rock, David Forsyth
University of Illinois at Urbana-Champaign
{bhattad2, jjrock2, daf}@illinois.edu

Abstract

Detecting anomalous faces has important applications. For example, a system might tell when a train driver is incapacitated by a medical event, and assist in adopting a safe recovery strategy. These applications are demanding, because they require accurate detection of rare anomalies that may be seen only at runtime. Such a setting causes building supervised methods unsuitable because of difficulty in collecting rare and generalizable anomaly data. We describe an unsupervised method for detecting an anomalous face image that meets these requirements. We present a novel feature construction technique that reliably has large entries for anomalous images, then use various simple unsupervised methods to score the image based on extracted features. Obvious constructions (autoencoder codes; autoencoder residuals) are defeated by a 'peeking' behavior in autoencoders. Our feature construction removes rectangular patches from the image, predicts the likely content of the patch conditioned on the rest of the image using a specially trained autoencoder, then compares the result to the image. High residual scores suggest that the patch was difficult for an autoencoder to predict, and so is likely anomalous. We demonstrate that our method can identify real anomalous face images in pools of typical (natural) images, taken from celeb-A, that is much larger than usual in state-of-the-art experiments. A control experiment based on our method with another set of normal celebrity images - a 'typical set', but non-celeb-A are not identified as anomalous; confirms this is not due to special properties of celeb-A.

1 Introduction

We describe a completely unsupervised method for detecting anomalous faces in images that do not require any example anomaly in training. Our method uses a novel representation of appearance, by extracting features from inpainting autoencoder residuals. We demonstrate that our method significantly improves over a number of natural baselines.

Detecting anomalous faces has important critical applications. For instance, a machine operator might fall asleep or a patient in an intensive care unit might have a heart attack. Ideally, a monitoring system would identify this sort of problem by watching the target's face and trigger some form of intervention. The crucial difficulty in building such a system is that there aren't datasets showing (say) people having heart attacks. Moreover, a reliable anomaly detection system must be built without seeing actual anomalies to generalize well.

This example presents serious difficulties for current methods for anomaly detection (briefly reviewed below), because previous anomaly detection systems tend to be evaluated on datasets where anomalies are very different from typical or natural (henceforth referred as typical) examples. But anomalous faces look quite similar to typical faces. Our method requires a representation of face appearance which exaggerates the relatively small changes that make a face image anomalous, without actually being shown. Worse, because face images are relatively high dimensional, there is no practical

Preprint. Work in progress.

prospect of simply applying a density estimator to the example images. Our strategy is to learn a compression procedure that reconstructs faces well, but not other similar unseen images, and then look at the residuals. This is not a routine application because one must be sure that (a) training images reconstruct well (routine) but (b) other similar anomalous images do not (tricky, and unusual). We show that a carefully designed residual of a specially trained, inpainting autoencoder has these two properties and therefore provides a strong feature extraction technique for identifying facial anomalies.

Contributions: We present a novel feature learning approach for anomaly detection using inpainting auto-encoders that do not require any anomaly images while training. We augment the Celeb-A dataset [19] for evaluating *true* image anomaly detection. We build a dataset of real anomalous faces and real typical faces to evaluate the proposed framework. We demonstrate that our feature works well in both supervised and unsupervised applications.

2 Background

Anomaly detection has widespread applications, including: image matting [12]; identifying cancerous tissue [1]; finding problems in textiles [24, 20]; and preventing face spoofing [2]. There is a recent survey in [6]. There are two distinct types of approach in the literature. In one approach, examples of both inliers and outliers are available, and discriminative procedures can be used to build representations and identify and select features. In the other, one can model only inliers, and anomalies are available only at test time.

Face anomaly detection is a good example problem because (a) data resources of typical faces are abundant and (b) anomalous faces look a lot like typical faces; trivial methods perform poorly. We do not assume that anomalous faces are available at training time, because doing so creates two problems. First, anomalous face images are rare (which is why they're anomalous) and aren't highly variable in appearance compared to typical image, so a dataset of reasonable size is difficult to build. Second, the estimate of the decision boundary produced by any particular set of anomalous face images is likely to be inaccurate. The location of the decision boundary is determined by both the anomalies and the typical images; but the anomalies must be severely undersampled, and so contribute significant variance to the estimate of the decision boundary.

Instead, we assume that only typical faces are available at training time. We must now build some form of distribution model for true faces and exploit it to tell how uncommon the current image is. We focus on building a feature construction that allows simple mechanisms to compute an anomaly score. An alternative is to use a kernel method to build a distribution model (the **one-class SVM** of [23]). We use this method as a baseline on autoencoder residuals, and when trained on our extracted features, we show that they outperform these baselines. [32] applies a Gaussian mixture model to the autoencoder residual and the code space to cluster them into two components. In their approach, they also feed in anomaly images. In our case, we define anomalies to be extremely rare and do not use any anomaly images while training. We believe for a better generalization on any unseen and variants of an anomaly, our approach is more desirable.

Our feature construction uses a specially trained **autoencoder** [13]. Auto encoders use an encoder to compress a signal to a code, which can then be decompressed. The code is a low dimensional representation of content which has been shown to be useul for tasks such as: appearance editing [28, 18]; inpainting [21]; and colorization [8]. Generative adversarial networks (GAN) [11] have been used for anomaly detection in [22], but one must build a distribution for the code. [22] do explore using residuals in combination with the code likelihood. However, because their model is built on a GAN, their inference procedure is quite expensive, requiring many backprop and gradient steps, while our method is simply a forward run through an autoencoder. Our model also introduces a novel inpainting conditioning strategy for feature construction.

Evaluating anomaly detectors is tricky, because anomalies are rare. One strategy is to regard one class of image as typical, and another as anomalous. This strategy is popular [30, 17, 7, 15] but may mislead. The danger is that one may unknowingly work with two very different classes, meaning that the quality of the distribution model for the typical class is not tested. In contrast, face anomaly detection has the advantage of being (a) intrinsically useful and (b) clearly difficult.

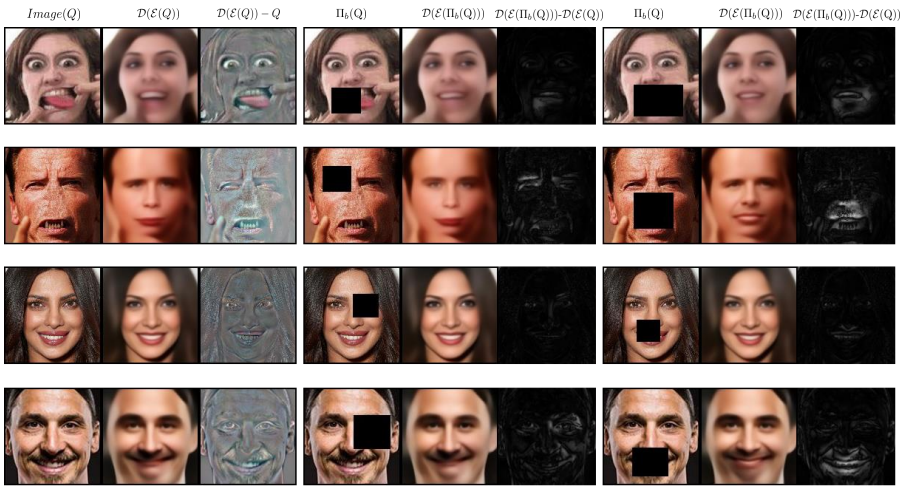

Figure 1: Forcing an autoencoder to inpaint at test time has important effects on the reconstruction. **Top two rows:** anomalous face images; **bottom two rows** typical face images. In the **first column**, the three images are actual input images, autoencoder's reconstructed images and the autoencoder residuals respectively. In the **second** and **third column**, the first image is the masked input, second is the autoencoder's reconstructed images and third is the residual difference between inpainted reconstruction residual from the residuals in the the first column. Notice how, for anomalous faces, not showing the autoencoder the content of the box affects the reconstruction. In the top row, attend to the dark bar at the left side of the model's mouth, significantly reduced when the autoencoder reconstructs without seeing Q (i.e. no peeking). Similarly, concealing the whole mouth results in a much more conventional reconstruction of the mouth. As a result, the residual error emphasizes where the image is anomalous. For the second row, note how eye size and gaze are affected; and the significant change in reconstructed mouth shape when the mouth is concealed. This effect is minor for typical faces. As a result, residuals against autoencoder inpainting are strong cues to anomaly.

The set building method of [29] could be applied to face anomaly detection. This approach has been shown to be accurate at identifying the one special face in a set of 16. A direct comparison is not possible, because their method relies on supervised inference, identifying the one different face in a set (i.e. given 15 smiling faces and one frowning face, it should mark the frowning face). However, we adopt their evaluation methodology and use analogous scoring methods.

3 Anomaly Features

We view anomaly detection as feature construction followed by a simple unsupervised method. Natural choices of feature constructions are autoencoder codes, pretrained discriminative models (eg [5]), or autoencoder residual features. An anomalous face image will look mostly like a typical face image, but will display some crucial differences. The problem is we don't know where those differences are or what they look like. A natural strategy is based on a generative models of typical face images. Write Q for a test image, and $\mathcal{M}(Q;\theta)$ for a learned model that produces the typical face image that is 'closest' to the query image. We could then use the difference $\mathcal{M}(Q;\theta) - Q$ to compute a score of anomaly. In practice, "peeking" by the learned model (details below) means that this approach fails. The learning procedure results in a model that is biased to produce a $\mathcal{M}(Q;\theta)$ that is closer to Q than it should be.

A simple variant of this approach is extremely effective. Rather than requiring $\mathcal{M}(.;\theta)$ to make the closest typical image, we conceal part of Q from $\mathcal{M}(.;\theta)$ and require it to extrapolate. We then compare the extrapolated region to Q to produce the anomaly signal.

3.1 Autoencoder Residuals as Anomaly Signals

We will build $\mathcal{M}$ using an autoencoder. Autoencoders construct low dimensional latent variable models from high dimensional signals. An encoder $\mathcal{E}$ estimates the latent variable (code; z) for a given input Q; a decoder $\mathcal{D}$ recovers the signal from that code. The two are trained together, using

criteria like the accuracy of the signal recovery (ie $|\mathcal{D}(\mathcal{E}(Q)) - Q|^2$; [3]). Variational versions which use Bayes priors on the code have been explored as well [16]. As we show in Figure 3, the code produced by the encoder is a poor guide to anomaly, likely because it is still fairly high in dimension, and an appropriate distribution model is obscure [22]. The autoencoder image reconstruction residual, $\mathcal{D}(\mathcal{E}(Q)) - Q$, is an alternative.

Straightforward experiments establish that the residual is a poor anomaly signal (Figure 1). The reason is interesting. An autoencoder is trained to reproduce signals from its training set, but this regime does not necessarily discourage reproducing other images as well. An autoencoder that is trained to reproduce face images accurately has not been trained *not* to reproduce (say) cat images accurately, too. This means the autoencoder could reduce the training loss by adopting a compression strategy that works for many kinds of images. Therefore, a compression procedure that is good at compressing face images is not necessarily bad at compressing other images. This problem is not confined to neural networks. For example, choice of principal components that represents face images well [25] may represent (say) cat images. Denoising in current implementations [26] does not cure this problem. For example, a good denoising strategy is to construct a large dictionary of patches, then report the closest patch to the input. While a dictionary built on faces may reproduce some classes of image poorly, there is no guarantee in the training loss. Requiring a 'small' code [14] or adding code regularization [16] does not cure this problem either, because it is not known how to account for the information content of the code. As a result, the model $\mathcal{M}$ built by the autoencoder is not guaranteed to report the typical face image that is 'closest' to the query image; instead, it may pass through some of the query image as well ('peeking' at the query image), so resulting in a small residual and a poor anomaly signal. Experimental experience suggests that neural networks quite reliably adopt unexpected strategies for minimizing loss ('cheating' during training), meaning that we expect peeking to occur, and Figure 1 confirms that it does. Peeking can be overcome by forcing the autoencoder to fill in moderately large holes in the query image.

4 Beating Peeking with Inpainting

Write Π_b for an operator that takes an image and overwrites a box b with zeros; write $\Pi_{\overline{b}}$ for an operator that overwrites all but the box b with zeros. These boxes will be quite large in practice. We will train an autoencoder $(\mathcal{E}, \mathcal{D})$ as below. We build an anomaly feature vector by constructing $|\Pi_{\overline{b}}[\mathcal{D}(\mathcal{E}(\Pi_b(Q))) - Q]\,|$ for a variety of boxes (as below). We will then apply simple decision procedures to this feature.

This feature works because the autoencoder cannot peek into Q within the box. Instead, it must extrapolate into b, and that extrapolation is difficult to produce for multiple classes. In turn, the extrapolate is a much better estimate of what a typical face image would look like within b, conditioned on the rest of Q. For example, if b spans a mouth, then the auto encoder constructs a typical mouth conditioned on the face and compares it with the observed mouth, and if the mouth is anomalous the residual will be large (Figure 1).

This is similar to the inpainting problem explored in [21].Our application of inpainting methods differs sharply from the usual in computer vision. First, we inpaint large blocks (trails and scratches are more usual, but see [21]). Second, the residual error in the inpainting is a feature. In contrast, image inpainting methods seek results that fill in plausible textures. In our method, small residuals are used to identify anomalous faces. This means that accuracy is important. For example, as our results show (Figure 2) methods like skip connections that increase the visual plausibility of an inpainted region in fact result in weaker anomaly detection. This is likely because these methods introduce high frequency details (so improving plausibility) but place them incorrectly (so increasing the residual). To the best of our knowledge, we haven't seen inpainting autoencoders used for extracting features in the literature. Our autoencoder is trained to inpaint randomly selected boxes. We use $|(\mathcal{D}(\mathcal{E}(\Pi_b(Q))) - Q)|_1$ as a training loss, thus requiring the autoencoder to inpaint. The only difference between this and a denoising regime is the size of the boxes, which is large compared to gaussian noise. At test, we use a fixed size box moved in a grid to extract features.

Our encoder and decoder use standard convolutional architectures with a fully connected layer for code construction. Average pooling is used for downsampling, and bilinear interpolation is used for upsampling. Following [4], we use a higher capacity network for the encoder than the decoder which seems to help with reconstructing higher frequency information. We use the elu non-linearity and

batch normalization after each conv layer, and a tanh non-linearity on the output from the decoder. We use the L_1 norm for our training loss.

5 Anomaly Experiment

We wish to determine whether we can detect true anomalies in a realistic setting. We use the Celeb-A dataset [19], which is a collection of thousands of labeled faces. As in [4], we filter and and also get crops of 128x128 with the Viola-Jones face detector [27], resulting in frontal faces in tightly cropped 128x128 boxes. After this step, we are left with less than 150k images of the total 200k in CelebA. We train our inpainting autoencoder with random Π_b. During test we use the same model to construct autoencoder codes as well as the inpainting features. For inpainting features, we use 32x32 boxes in a regular grid. We exclude the boxes that would lie directly on the image boundary. For Resnet features we use a pretrained resnet trained on face recognition from [5], we remove the final softmax layer, and use the resulting network as a feature constructor. We must also note that the pretrained resnet was trained on very large pool of the face dataset. There is a plausibility of leaking of the validation and test features into the model.

However rather than considering any specific attribute, we consider the entire Celeb-A dataset as typical data. We set aside 20,000 images for use in test and 10000 for validation, leaving us with 125,253 images for training. For our anomaly images, we collected a 316 image anomaly dataset. This set, which we call the **anomaly set** is comprised of strange or "weird" faces. It includes extreme makeup, masks, photoshops, and people making extreme faces. We pass the images through the same Viola-Jones detector and cropper. This rejects many of the anomaly images, and in fact we only have about a 10% yield on anomaly images, meaning that we had to find roughly 3000 anomaly images in order to get 316. However, this construction is sensible: if Viola-Jones does not believe the images have a face, then they are too obviously anomalous. For example, a photograph of a cat would have a high anomaly score under our method, but a cat is also not likely to be identified as a face by a competent detector, so determining it is an anomaly is not particularly important or difficult.

We also wish to determine if anomaly detection is caused by special features of the Celeb-A dataset. An anomaly detector which identifies any image not from Celeb-A would not be particularly useful. We therefore collect a **typical set** of 100 images we do not believe to be anomalous images. It is comprised of pictures of celebrities that were taken after Celeb-A was created so there are no overlaps in pictures. We also tried to find new celebrities, so that the people would be less likely to have appeared in the original Celeb-A dataset. This dataset is used to validate that a method is not memorizing images in Celeb-A or finding a particular feature of Celeb-A and rejecting any new images. We show samples from Celeb-A, the typical set and the anomaly set in figure in the supplementary. By example it is reasonable to ask an anomaly detection method to identify images from the anomaly set without identifying images from the typical set.

Our experiments are modeled on the set experiment presented in [29]. They form a set of 16 images from Celeb-A where 15 images share at two attributes, and one image differs. The goal is to identify the image with different attributes in a supervised setting. We adjust this slightly. Large sets are more indicative of performance for real world anomaly detection, where the goal is to identify one image in thousands rather than one image in ten. However, using large sets is significantly more difficult so we report recall at 1, 5, and 10 rather than just reporting recall at 1. Note that at no point does any method have access to labels, which are revealed only to evaluate the experiment. We believe this is a better model for detecting rare anomalies.

Evaluating anomaly detection: We select one image from the anomaly set, and between 15 and 299 images from the 20,000 celeb-A held out images (without consideration of attributes, in contrast to [29]). We then score each image using our feature and a variety of scoring methods (Section 5.1) to evaluate recall for the anomaly image, averaged over 10,000 sets. As figure 3 shows, recall is strong even from large sets, and the choice of score appears not to matter.

Control: Strong results could be caused by some special feature of celeb-A images. To control for this possibility, we repeat the anomaly detection experiment, but replacing the image from the anomaly set with an image from the typical set (100 typical images not from celeb-A). If celeb-A were wholly representative, then this experiment should produce recalls at chance. A figure in supplementary shows, the results are not at chance (there is something interesting lurking in celeb-A),

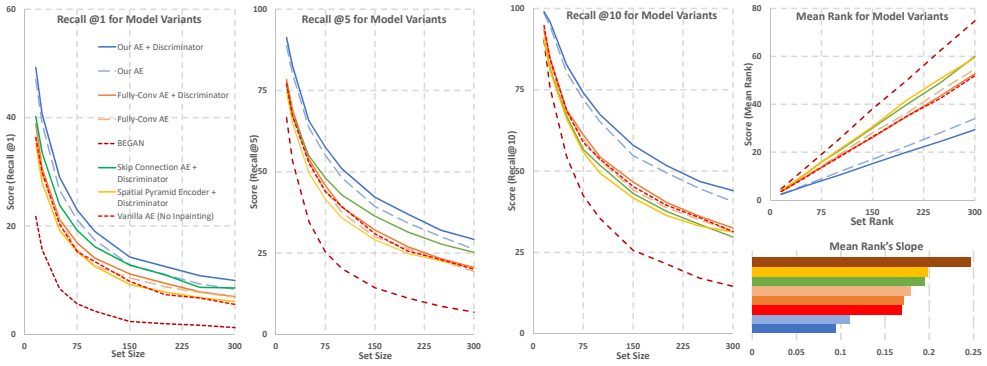

Figure 2: Recall at 1, 5 10 and mean rank for various scores using different models. For recalls, higher the better and for the mean rank, lower the better. We find that the network design is critical for the face anomaly detection. The models that do well for inpainting; for instance using skip connections, need not improve anomaly detection score, rather they make them worse. GAN's perform worse. They require a search for latent codes (1000 backpropagation steps - as described in Section 2). Recalls are averaged over 10000 trials, and so have very low variance.

but recall is very much weaker than for anomalies. The performance of the anomaly detector cannot be explained by quirks of celeb-A

Hardness: It is possible that collection methods, etc. mean that the anomalous images have (say) backgrounds that are different from Celeb-A images, and so that anomaly detection uses entirely the wrong cues. We calibrate all images as to hardness with the following procedure. We compute residual features for all images. We then repeatedly separate out a small training set of both 100 anomalous and 100 non-anomalous images, fit a linear SVM, and record for each other image whether it was correctly classified (1) or not (0). The average of this score yields a hardness measure, where low values indicate that, with reasonable features and a reasonably trained classifier, the image could not be classified correctly.

5.1 Unsupervised Feature Learning

We use our regular grid of residual features for 32x32 patches with a 32 pixel edge exclusion and explore a variety of methods for turning the residual features into an anomaly score. The mean over the feature vector makes up our main method due to its simplicity and good performance. For our feature, we also found that the L_∞ norm finds the most violated residual from the set of patches, which is obviously useful for anomalies that tend to occur locally. For the **Adversarial Losses,** we use the idea from [31] and add a patch GAN like discriminator to differentiate between the inpainted image features and the original image features. We deploy two discriminators, one for the patch that is inpainted and the other for the entire image. We found maximizing the perceptual distance using [9] to give us the best performance.

The **Mahalanobis Distance** (mahal) estimates a mean and covariance from a set and then measures distance with respect to the mean and covariance. It is typical to estimate the mean and covariance on training data.The **Equivariant Transform** (equivariant) introduced in [29] can be applied in an unsupervised manner on a set of images. A sensible version looks like the Mahalanobis Distance. Recall the equivariant transform in matrix form:

$$\mathbf{X} = \left[\lambda \mathbf{I} + \gamma(\mathbf{1}\mathbf{1}^{\mathbf{T}})\right] \mathbf{X} \tag{1}$$

Which for an element x_i is equivalent to

$$\hat{x_i} = \lambda x_i + \gamma \sum_i x_i \tag{2}$$

Let Σ^{-1} be the inverse covariance of $\mathbf{X}$, then $\gamma = -\Sigma^{-1/2}/N$ and $\lambda = \Sigma^{-1/2}$ compute a transformation which under the L_2 is the Mahalanobis Distance. This transformation is sensible and can be applied to the data prior to applying the mean to reweight the feature dimensions and take into

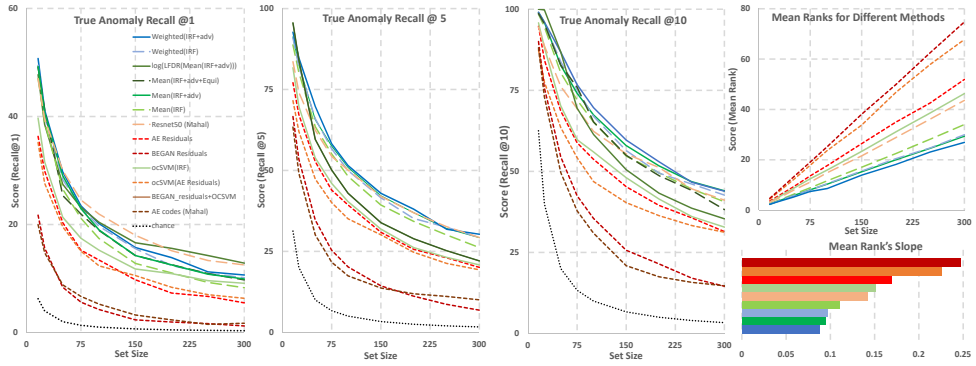

Figure 3: Recall at 1, 5 and 10 for various scores using our anomaly feature plotted against the size of the set from which the anomaly must be picked. We also show mean rank plot for all methods against the size of the set. For recalls, higher the better and for the mean rank, lower the better. Here, we show the performance on our new data set on the autoencoders' inpainted residual as features (IRF), and by applying various transformations to our features(ocSVM, equivariant transformation (equi), adversarial losses (adv), log(lfdr)). We use autoencoder residuals, BEGAN residuals and features drawn from Resnet50 and autoencoder's code as our baseline. Recall at 1, 5 and 10 are hard for all methods and beat chance by a huge margin. Note also the test is demanding compared to the literature; a single anomalous face must be picked from up to 300 others. However, these results might depend on some signal property of the celeb-A dataset. The mean plot captures the overall effectiveness of the model and the features. Also, on the recall plots, pretrained Resnet50 looks to be a strong competitor to our method, but from the mean rank plot it is evident that Resnet50 perform significantly worse on many images. Adversarial losses improves our results by $2-3\%$ and other transformations does not have any major effect to the performance, implying the robustness of the IRF. A similar plot in supplementary shows results from our control experiment, where the image used as an anomalous image is a typical face image that doesn't appear in celeb-A (details in section 5). Recalls are averaged over 10000 trials, and so have very low variance. (Legends are same for each sub-plot)

account that some dimensions might be highly varying while others are not. For our autoencoder residual feature, we assume that our features are IID, so we can estimate a diagonal covariance, and we compute a robust mean and covariance by eliminating the largest and smallest values on each feature. Note that this is done without knowing which item is anomalous and thus does not violate train-test splits.

The **Local False Discovery Rate** (lfdr) is a construction that identifies the probability that an item comes from a null distribution, without knowing what the null is [10]. The method originates in multiple hypothesis testing, assuming that most observations come from the null. Assume the null distribution is $f_o(z)$, the non-null is $f_1(z)$, and the prior an item comes from the null is π_o. Then the lfdr is

$$p(\text{null}|z) = \frac{\pi_o f_o(z)}{\pi_o f_o(z) + (1-\pi_o) f_1(z)} \tag{3}$$

Small values suggest an item is worth investigating (i.e., anomalous). Estimation is complicated by the fact that neither $f_o(z)$ nor $f_1(z)$ are known; but the assumption that π_o is large, and $f_o(z)$ is 'close' to a standard normal distribution allows fairly accurate estimation. We used the R program `locfdr`. We estimated local false discovery rates using all 20416 test data items (doing so does not involve knowing which item is anomalous, so does not violate test-train protocols). We found the estimate a standardized version of the log of the mean of inpainting residual features to work best for lfdr.

5.2 Results

As seen in figure 3, our feature performs well regardless of feature transformation applied. We identify **anomalies** at rates significantly greater than chance even as the size of the set increases. Resnet-50 features [5] with a Mahalanobis Distance represents a strong baseline, however, we outperform it regardless of the possibility of leaking of test CelebA featurs. There does seem to be some bias in the

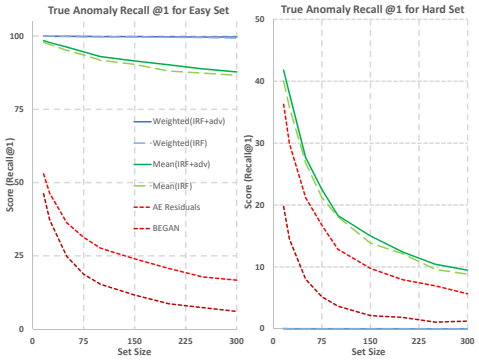

Figure 4: Hardness is defined as in section 5.Here we show recall at 1 for the easy set, where the minimum hardness measure over all images in each test set is 1 (meaning that hardest is still easily distinguishable) and for a hard set (where the minimum hardness is 5). Note how the method works well on easy, but collapses completely on hard, suggesting its effectiveness is effect of bias. All methods find hard data hard.

Celeb-A dataset being used to identify anomalies but our features and the Resnet-50 features do not identify **typical** images at anywhere near the same rate as anomalies. The gap between performance on typical images and anomalous images is apparent and clearly significant (eg. 50 vs 20 percent for recall at 1 in a 16 image set).

Weights: Anomalous face detection is an experimentally delicate problem, where dataset bias issues can be significant. Here is a simple method that outperforms our method. For each block in the 32x32 grid, compute how often that block has the largest residual. Now weight the residual at that block by this frequency weight. As Figure 3 indicates, the method is very effective. However, this is an artifact of dataset bias. The method in fact is identifying background blocks (whose residual tends to be large), and emphasizing them in the score. In turn, this exploits the difficulty in matching backgrounds in collected anomalies with Celeb-A images. One can see this by looking at Figure 4; here results are shown for data with easy hardness measure (where the weighted method does very well indeed, because such anomalies tend to have backgrounds very different to the data) and with hard hardness measure (where it collapses completely because the background cue is absent).

Limitation: A limitation of our approach is that inpainting helps in detecting concentrated anomalies quite well as compared to the diffused ones. An ideal strategy to overcome this would be inpainting multiple boxes instead of one or conditioning on a smaller visible box to generate entire face image. This strategy would then perform poorly on concentrated anomalies. But an ensemble approach can be adopted to get improved performance on both concentrated and diffused anomalies.

6 Conclusion

For the first time, we demonstrate the inpainting autoencoder residuals as a feature for combating the overgeneralization of compression losses. This allows us to train our method solely on non-anomalous data, mimicking how a real anomaly detector must be trained. We demonstrate that our inpainting residual features are useful and work well in unsupervised settings. Though we did not see improvement in performance, it is easy to use inpainting autoencoder features with various feature transformation techniques. We also describe a standard anomaly detection experiment for evaluating future anomaly work on image sets, enabled through the collected two small datasets to augment Celeb-A.

References

[1] Sharon Alpert and Pavel Kisilev. Unsupervised detection of abnormalities in medical images using salient features. In Sebastien Ourselin and Martin A Styner, editors, *SPIE Medical Imaging*, pages 903416–7. SPIE, March 2014.

[2] Shervin Rahimzadeh Arashloo, Josef Kittler, and William Christmas. An Anomaly Detection Approach to Face Spoofing Detection: A New Formulation and Evaluation Protocol. *IEEE Access*, 5:13868–13882, 2017.

[3] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. *arXiv preprint arXiv:1710.08092*, 2017.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *Acm Computing Surveys*, 41(3), 2009.

[7] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Anomaly detection with generative adversarial networks, 2018.

[8] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Forsyth. Learning Diverse Image Colorization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2877–2885. IEEE, 2017.

[9] Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative modeling using the sliced wasserstein distance. *arXiv preprint arXiv:1803.11188*, 2018.

[10] Bradley Efron. Size, power and false discovery rates. *Ann. Statist.*, 35(4):1351–1377, 08 2007.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mirza Mehdi, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. In *NIPS*, June 2014.

[12] D Hasler, L Sbaiz, S Susstrunk, and M Vetterli. Outlier modeling in image matching. *IEEE TPAMI*, 25(3):301–315, March 2003.

[13] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[14] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994.

[15] Matthias Schubert Jindong Gu and Volker Tresp. Semi-supervised outlier detection using generative and adversary framework, 2018.

[16] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

[17] Mark Kliger and Shachar Fleishman. Novelty detection with GAN, 2018.

[18] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and MarcAurelio Ronzato. Fader Networks: Manipulating Images by Sliding Attributes. *arXiv.org*, pages 1–10, June 2017.

[19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[20] K L Mak, P Peng, and H Y K Lau. A real-time computer vision system for detecting defects in textile fabrics. In *2005 IEEE International Conference on Industrial Technology*, pages 469–474. IEEE, 2005.

[21] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature Learning by Inpainting. In *Computer Vision and Pattern Recognition*, April 2016.

[22] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[23] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[24] A Serdaroglu, A Ertuzun, and A Ercil. Defect detection in textile fabric images using wavelet transforms and independent component analysis. *Pattern Recognition and Image Analysis*, 16(1):61–64, 2006.

[25] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.

[26] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *The Journal of Machine Learning Research*, 11:3371–3408, December 2010.

[27] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, volume 1. IEEE, 2001.

[28] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2Image: Conditional Image Generation from Visual Attributes. In *European Conference on Computer Vision*, 2016.

[29] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep Sets. In *NIPS*, pages 3394–3404, 2017.

[30] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, pages 1100–1109, 2016.

[31] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[32] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection.