FINGERPRINTS OF SUPER-RESOLUTION NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Several recent studies have demonstrated that deep-learning based image generation models, such as GANs, can be uniquely identified, and possibly even reverse engineered, by the fingerprints they leave on their output images. We extend this research to a previously unstudied type of image generator: single image super-resolution (SISR) networks. Compared to previously studied models, SISR networks are a uniquely challenging class of image generation model from which to extract and analyze fingerprints, as they can often generate images that closely match the corresponding ground truth and thus likely leave little flexibility to embed signatures. We take SISR models as examples to investigate if the findings from the previous work on fingerprints of GAN-based networks are valid for general image generation models. In this paper, we present an analysis of the capabilities and limitations of model fingerprinting in this domain. We show that SISR networks with a high upscaling factor or trained using adversarial loss leave highly distinctive fingerprints, and show promising results for reverse engineering some hyperparameters of SISR networks, including scale and loss function.

1 INTRODUCTION

Recent progress in deep-learning based image synthesis has dramatically reduced the effort needed to produce realistic but fake images (Tolosana et al., 2020). But just as a criminal may leave fingerprints at a crime scene, image synthesis networks leave telltale "fingerprints" on the images they generate (Marra et al., 2019). Researchers have sought to extract increasingly detailed information about image synthesis networks from these fingerprints. A popular form of this problem is *deepfake detection* (Dolhansky et al., 2019), which seeks to extract a single bit of information: is a particular image real or fake? Going further, *model attribution* seeks to identify the particular image generation model that produced an image (Yu et al., 2019). *Model parsing* goes even further, seeking to infer design details of the image generation model (Asnani et al., 2021).

Model fingerprints have been studied primarily to identify and track down sources of misinformation. Therefore, the types of models used for deepfakes, such as generative adversarial networks (GANs), have received the most attention. But model fingerprinting has other applications, such as intellectual property protection. A proprietor may have property rights to a dataset, loss function, neural network architecture, or pretrained model which is useful for image synthesis. Such a proprietor might be interested in identifying media artefacts created in violation of their intellectual property rights. In this situation, model attribution can detect the misuse of pretrained networks, and model parsing is useful for detecting when individual parts of an image synthesis network (e.g. the training dataset, the loss function) are proprietary. Proprietors may wish to deploy this kind of copyright detector for a broad range of deep image generation models. For example, deep-learning based image and video enhancement technologies, such as super-resolution, image de-noising, and video interpolation are all commercially important applications a proprietor may wish to protect.

Without further study, it is unclear how the current, GAN-focused model fingerprint literature will extend to a broader set of image generation models. Compared to GANs, image enhancement networks often produce images which are very close to the ground truth training targets. For example, different L_1-optimized single image super-resolution (SISR) models are known to converge on super-resolved outputs which are visually very similar (Sajjadi et al., 2017). *Are such subtle variations between models sufficient to produce the uniquely identifying fingerprints?*

As an initial foray into the study of fingerprints for more general image generation models, we study the model fingerprints for SISR networks. We collect photographs from Flickr and super-resolve each of them with 124 different SISR models. These 124 models consist of 16 pretrained models published online by other researchers, and 108 models which we have trained ourselves by systematically varying the architecture, super-resolution scale, training dataset, and loss function. We then train an extensive collection of image classifiers to perform model attribution, and to predict four important hyperparameters of each SISR model: architecture, dataset, scale, and loss function. By systematically reserving different subsets of the SISR models for testing, we explore the generalization capability of our model attribution and parsing classifiers. Our contributions are as follows:

- We develop a novel dataset of 124 super-resolution models, which will be made publicly available.
- We analyze the factors that contribute to the distinctiveness of a SISR model fingerprint. We show that the choice of scaling factor and loss function significantly impact fingerprint uniqueness.
- As Yu et al. (2019) showed for GANs, we show that the fingerprints of adversarially-optimized SISR models are highly sensitive to small changes in hyperparameters, such as random seed.
- We study the generalization of our SISR model attribution classifier to models outside the training set. We show that our attribution classifier generalizes well from our contrived training set to real-world models, with architectures and loss functions not seen during training.
- We train a set of model parsing classifiers to predict the hyperparameters of the SISR models. We show promising results for predicting the scale and loss function, and mixed results for the architecture and training dataset.

2 Related Work

Single image super-resolution: Recent years have seen rapid progress in deep-learning based SISR methods. These days, there are a profusion of such methods available online. We choose SISR methods as our subject of study for this reason. A diverse set of state-of-the-art SISR models form the foundation of our experiments. As listed in Table 4 in the appendix, we selected 12 SISR methods presented in recent papers based on their reproducibility and their high-quality results.

Model attribution: To identify the source of synthetic images, model attribution methods can look either for watermarks (signatures deliberately encoded into each output image by the network author) (Hayes et al., 2020; Skripniuk et al., 2020; Adi et al., 2018; Yu et al., 2020; Zhang et al., 2020), or for unintentional statistical anomalies in the generated images which are unique to a particular model, which we call "fingerprints". We focus on detecting these fingerprints, which can appear without deliberate intervention by the network author.

Generative adversarial networks have been shown to possess uniquely identifying fingerprints (Marra et al., 2019; Yu et al., 2019). These fingerprints have been extracted with convolutional networks (Yu et al., 2019; Xuan et al., 2019), and with hand-crafted features (Marra et al., 2019; Guarnera et al., 2020; Goebel et al., 2020). Albright & McCloskey (2019) and Zhang et al. (2021) showed that GAN inversion can be a useful way to attribute images to the GANs. All of these methods focus on finding the fingerprints of GANs or Variational Auto Encoders (VAEs). However, the relevance of these results to image enhancement models, such as SISR networks, is a priori unclear.

Reverse engineering/model parsing: We are the first to attempt to reverse engineer some hyperparameters of SISR models from the images they generate. But we are not the first to attempt to reverse engineer the hyperparameters of black-box neural networks. By feeding around 100 specially-designed input images through a black-box classifier, Oh et al. (2019) successfully inferred many fine-grained architectural choices for small image classification networks. These input images were optimized using gradient descent so that the classifiers' output would correlate with its hyperparameters. Hyperparameters inferred included the network's activation function, optimization algorithm, and training dataset. By comparison, our experiments predict fewer hyperparameters of much larger networks, and do so by viewing arbitrary output images, without the need to pass specially-designed inputs through the black-box model.

In a concurrent work, Asnani et al. (2021) train a convolutional network to extract a fingerprint from a generated image, and to predict the hyperparameters of various image generation models from this fingerprint. Their method, like ours, can be used for attribution,and model parsing. Their

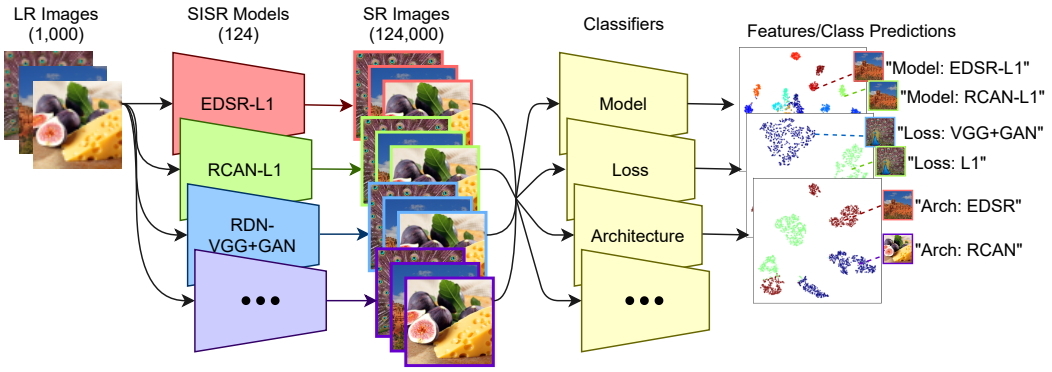


Figure 1: An overview of our experimental setup. 1,000 low-resolution images are fed through each of 124 SISR models to produce a dataset of 124,000 super-resolved images. We then train our model parsing and attribution classifiers on this dataset.

study covers a diverse domain of 100 image generation models, mostly unconditional GANs and variational autoencoders. Our work, by contrast, is focused on SISR models, which generate images very close to a ground truth and likely leave less flexibility to embed fingerprints.

3 Setup

We wish to explore what choices of SISR model hyperparameters lead to distinctive model fingerprints, and which model hyperparameters can be reverse engineered using model parsing. To explore these questions carefully, we need a dataset of SISR models that varies the hyperparameters we wish to experiment on, while holding all other hyperparameters constant. We choose 5 SISR model hyperparameters to analyze: model architecture, super-resolution scale factor, loss function, training dataset, and random seed. We train a 108 SISR models with various combinations of these experimental hyperparameters, and add in 16 more SISR models from previous works for additional diversity. We construct our main dataset by super-resolving 1,000 images by each of our 124 SISR models. We use this dataset to train several image classifiers for model attribution and parsing tasks. Finally, we can address our research questions by analyzing the performance of these classifiers. Figure 1 shows an overview of our process.

3.1 Single Image Super-Resolution Model Dataset

We want our collection of SISR models to meet the following criteria:

1. **Realistic**: Our SISR models should be comparable to those encountered in the real world.
2. **Diverse**: Our models should span many architectures, loss functions, and training sets.
3. **Large**: We want a large number of SISR models, so that the model classifiers can begin to generalize across the SISR model space.
4. **Uniform**: The hyperparameters of our SISR models should be uniformly distributed and independent from each other to prevent spurious correlations that could confound our analysis.

To make our dataset realistic and diverse, we include 16 real-world pretrained super-resolution models, published between 2016 and 2021 (See Table 4 in the appendix). Unfortunately, there are not enough pretrained SISR models available online to make our dataset very large. But even more importantly, distribution of hyperparameter values among these 16 SISR models is neither uniform nor independent. For example, all four models with 2x super-resolution scale are L_1-optimized. Such correlations are confounding factors in our analysis, and we would prefer to avoid them.

Therefore, we make our dataset large and uniform by adding 108 SISR models which we train ourselves. We select 5 SISR model hyperparameters to vary in our experiments (we will call these the *experimental hyperparameters*). All other training details of our custom-trained SISR models are held constant. The experimental hyperparameters and their values are as follows:

1. **Architecture**: The choices are **EDSR** (Lim et al., 2017), **RDN** (Zhang et al., 2018b), and **RCAN** (Zhang et al., 2018a).
2. **Dataset**: The super-resolution dataset used for training the model. The choices are **DIV2K** (Agustsson & Timofte, 2017) or **Flickr2K**, originally collected by Lim et al. (2017). To see if using a smaller training dataset might lead to a more distinctive model fingerprint, we also trained SISR models with just one quarter of the total training data available from these two datasets, effectively creating two more dataset choices, $\frac{1}{4}$**DIV2K** and $\frac{1}{4}$**Flickr2K**
3. **Scale**: Scaling factor by which to upsample the low-resolution input image; either **2x** or **4x**. To be clear, this is the scaling factor for the linear dimension of the image, not the total number of pixels; a 2x-upsampled image has four times as many pixels.
4. **Loss**: Loss function to optimize during training. Choices are the $\boldsymbol{L_1}$ **norm** (which is standard in the super-resolution literature), **VGG+adv.** loss, or **ResNet+adv.** loss. VGG+adv. loss is the the same linear combination of VGG-based perceptual loss and adversarial loss that was used in SRGAN (Wang et al., 2018a). ResNet+adv. uses the same adversarial term, but replaces the VGG-based perceptual loss with ResNet based perceptual loss (for details, see Appendix A.3).
5. **Seed**: Random seed to use during training. Random seeds used are simply **1**, **2**, or **3**.

In total, there are 216 possible SISR models that could be trained from different combinations of these hyperparameters. To save time and computational resources, we only train the subset of these models whose random seed is 1 *or* whose training dataset is DIV2K. This leaves us with 108 custom-trained SISR models.

3.2 Image Datasets

As discussed in Section 3.1, We employ two existing super-resolution image datasets, DIV2K and Flickr2K, to train our SISR models. We also create our own dataset of super-resolved images which we will use to train the model attribution and parsing classifiers. Because models often behave anomalously well on the images they were trained on, we do not wish to reuse any of the images from the DIV2K or Flickr2K datasets for our super-resolution dataset.

Instead, we collect a new image dataset consisting of 1,000 photographs from Flickr. We query Flickr for 200 images from each of the following 5 image tags: food, architecture, people, animals, and landscapes. We select only images with at least two megapixel resolution. At full resolution, however, many of these images contain visible JPEG artifacts, so we downsample the images to 960 pixels in their largest dimension, at which point any jpeg artifacts were imperceptible to us. We refer to this collection of images as the "Flickr1K dataset". Table 5 in the appendix compares some aggregate image statistics of this dataset to the DIV2K and Flickr2K datasets.

To generate the final super-resolution dataset upon which we train our model attribution and parsing classifiers, we super-resolve each image in our Flickr1K dataset by each of our 124 SISR models. For each SISR model, for each Flickr image, we first downsample the image by the model's scaling factor using bicubic interpolation, and then super-resolve the downsampled image using the model. This gives us a dataset of 124,000 super-resolved images. To get a sense of the resulting images, see Table 1, which displays the same 64x64 image patch super-resolved by 34 of the 124 SISR models.

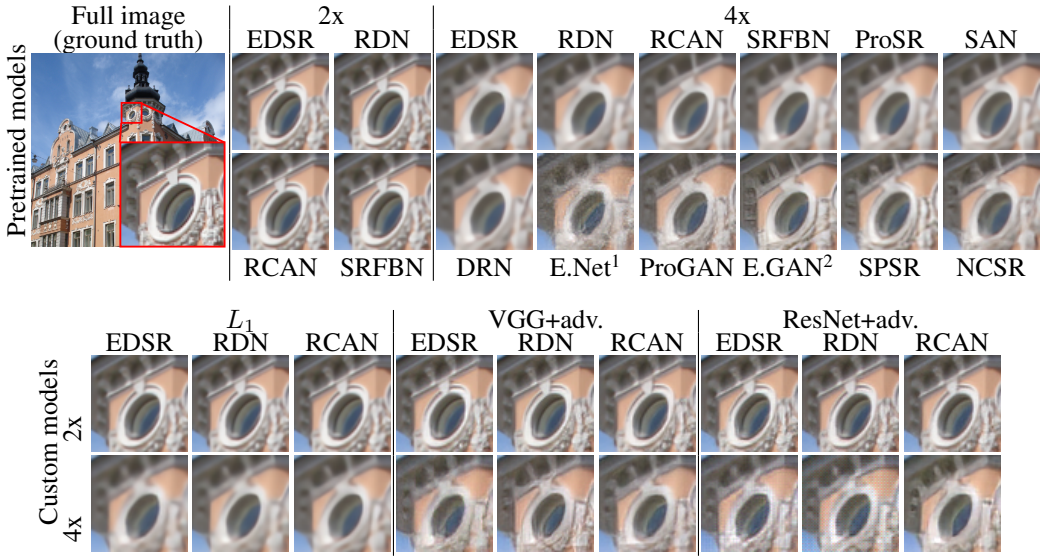
3.3 Classification Networks

We train an extensive collection of model classification networks to perform model attribution and parsing. Following Rössler et al. (2019), we choose XceptionNet, pretrained on ImageNet, as our backbone classification network of choice.

To adapt the pretrained XceptionNet to our classification problems, we replace the final fully-connected layer of the network with a randomly initialized one of the appropriate shape to output the right number of classes. A softmax layer is applied after the last fully-connected layer, and the network is trained with cross-entropy loss. As in Rössler et al. (2019), We train just the final layer for three epochs, then we unfreeze all network weights and fine-tune the entire network for 15

[1] EnhanceNet
[2] ESRGAN

Table 1: A small image patch super-resolved by the 16 pretrained SISR models (top) and a sample of 18 custom-trained SISR models (bottom). All custom-trained models shown were trained with the DIV2K dataset, with random seed 1. Best viewed zoomed in.



epochs. Our classifiers are trained with our super-resolved image dataset on just 800 images from each SISR model. We reserve an additional 100 images for validation, and 100 for testing. All analyses presented in Section 4 are computed from this test set of 100 super-resolved images per SISR model.

4 Experiments

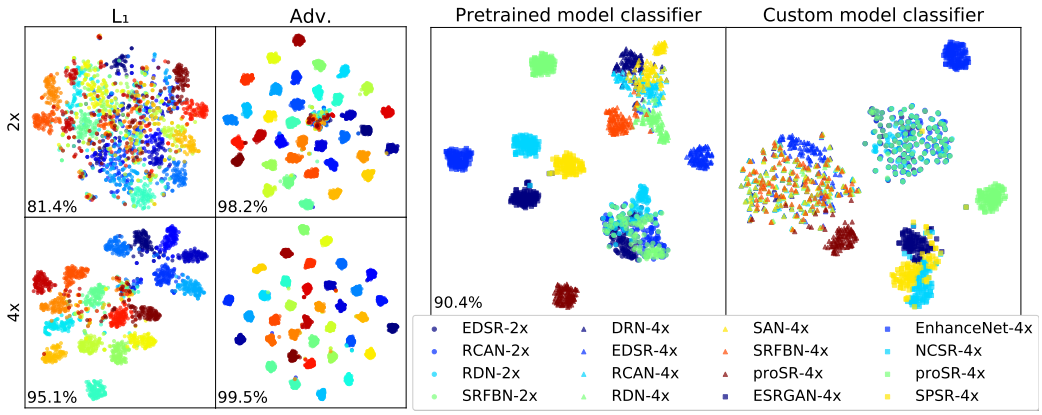
Our experiments are organized around the analysis of two different problems: model attribution (Section 4.1) and model parsing (Section 4.2). We formulate both as classification problems, and train XceptionNet models to solve them as described in Section 3.3.

4.1 Model Attribution

How reliably can a SISR model be uniquely identified by its output images? What combinations of hyperparameters lead to distinct fingerprints? To answer these questions, we train and analyze two attribution classifiers: the *custom model attribution classifier*, which is trained to distinguish between the 108 custom-trained SISR models, and the *pretrained model attribution classifier*, which is trained to distinguish between the 16 pretrained models. We discussed the comparative benefits and drawbacks of these two subsets of models in Section 3.1. Essentially, the the custom models are a larger and more controlled sample, while the pretrained models are more diverse. For the rest of the section, we use "model" as shorthand for "SISR model" unless explicitly stated otherwise.

4.1.1 When are SISR Model Fingerprints Distinctive?

Do certain hyperparameter choices make SISR model fingerprints more or less distinctive? We hypothesize that the more detail a SISR model adds to an image, the more opportunities there are for the model to embed its distinctive biases, i.e. to "leave fingerprints". 4x upscaling adds more detail than 2x. As noted in Sajjadi et al. (2017), L_1-optimized SISR models tend to converge upon the same unnaturally smooth super-resolved images, while SR images from adversarially trained models are more detailed and diverse. Therefore, *we hypothesize that 4x SISR models leave more distinctive fingerprints than 2x, and adversarially trained models leave more distinctive fingerprints than L_1 models*.



(a) Custom model image features (b) Pretrained model image features

Figure 2: T-SNE visualizations of of super-resolved image feature embeddings. Figure 2a shows the features of images upscaled by the 108 custom-trained models, grouped by scale and loss. Figure 2b shows feature embeddings for the pretrained SISR model images as encoded by the pretrained model attribution classifier (left) and the custom model attribution classifier (right). Classification accuracies associated with the models in each plot are in the lower-left corner.

To test this hypothesis, we look at the accuracy of our custom model attribution classifier segmented by each hyperparameter, as shown in Table 2. Attribution accuracy is roughly the same for the different training datasets and architectures, indicating that distinctiveness of SISR models does not vary much with these parameters. But classification accuracy varies significantly by scale and loss function: average classification accuracy for 4x SISR models is 5.4% higher than for 2x models, and average classification accuracy for adversarially trained models is 11.0% higher than for L_1 models.

Table 2: Accuracy (%) of our custom model attribution classifier grouped by different hyperparameters. For example, the average accuracy of our classifier on SISR models whose *scale* is *2x* is 92.6%.

	Value	Acc.
scale	2x	92.6
	4x	98.0
loss	L_1	88.2
	VGG+adv.	98.5
	ResNet+adv.	99.2
arch.	EDSR	95.7
	RDN	96.0
	RCAN	94.2
dataset	DIV2K	95.3
	Flickr2K	94.6
	$\frac{1}{4}$ DIV2K	95.3
	$\frac{1}{4}$ Flickr2K	96.1

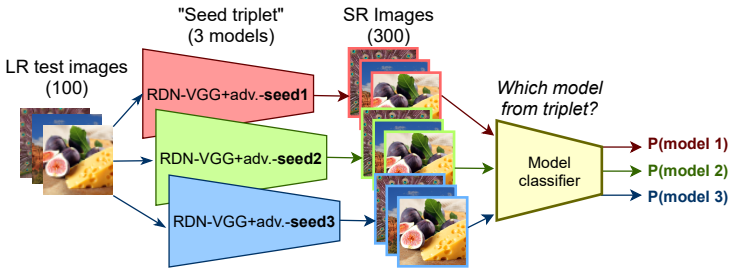


Figure 3: Distinguishing between models which differ only by seed.

Figure 2a shows a T-SNE embedding of the super-resolved image features disaggregated by scale and loss. We define *image features* as the 2048-dimensional vector of activations from the last layer of an attribution classifier. Class separation is better for 4x SISR models than 2x, and better for adversarial than L_1 models. *This data supports our hypothesis that adversarial loss functions and higher super-resolution scales lead to more distinctive model fingerprints.*

In Yu et al. (2019), even small variations in a GAN's training parameters can lead to highly distinctive GAN fingerprints. To test if this finding holds for our dataset, we evaluate the accuracy of our custom model attribution classifier at distinguishing between groups of models which differ only by their random seed (as shown in Figure 3). Our custom-trained SISR model dataset contains 36 "seed triplets": sets of three SISR models which are identical except for their seed. Our custom-trained model classifier can distinguish between models in each triplet with an average of 98.3% accuracy. Disaggregating by scale and loss shows the same trend as the broader attribution problem: the (2x, L_1), (4x, L_1), (2x, adv.), and (2x, adv.) model groups have 92.9%, 98.6%, 99.3%, and 99.9% accu-

racy on this task, respectively. We interpret this as a confirmation that Yu, et al.'s finding extends to this new domain.

4.1.2 PRETRAINED MODEL ATTRIBUTION

Our 16 pretrained models have a greater diversity of architectures, loss functions, and datasets, and may be more representative of the kinds of SISR models encountered "in the wild". So do our attribution results still hold for them? To test this, we train a model attribution classifier to predict which of the 16 pretrained models produced a given super-resolved image. We find that the performance of this classifier improves significantly if we initialize it with weights from the custom model attribution classifier, instead of starting from the XceptionNet model used for ImageNet classification.

Overall test-set attribution accuracy for these 16 pretrained models is 90.4%. Our hypothesis about scale and loss function still holds: accuracy among the (2x, L_1) models is 75.5%, while accuracy among the (4x, L_1) is 92.4%. The average classification accuracy for the (4x, adv.) group is 99.6%. Figure 2b shows a T-SNE embedding of the features of the test images classified by the pretrained model attribution classifier. The figure depicts a similar trend to Figure 2a: adversarially trained models are highly separable, (4x, L_1) models less so, and (2x, L_1) least separable of all.

4.1.3 FINGERPRINTING UNSEEN SISR MODELS

So far, we have assumed that the full set of SISR models our attribution classifiers will ever encounter is known and available during training. In real-world applications, such as scraping the web for super-resolved images which make illicit use of a proprietary model, this condition is unlikely to obtain. More likely, attribution classifiers will be met with images from numerous unknown sources, and will need to handle them gracefully. So does our attribution classifier still detect meaningful fingerprints for these unseen models?

To answer these questions, we ran the super-resolved images from our 16 pretrained models through the custom model attribution classifier, which has only seen our 108 custom-trained models during training. A T-SNE embedding of the resulting image features is displayed in Figure 2b. Notice that these features, taken from the custom attribution classifier, follow a very similar trend to those from the pretrained attribution classifier, which *has* seen these particular SISR models. Class separation is not as good, but the (4x, adv.) models are still modestly separable.

4.2 MODEL PARSING

We have demonstrated that small variations in SISR model parameters can lead to distinctive fingerprints. *Are these variations in model fingerprints random, or do they contain information about the underlying model parameters?* If they contain such information, this could be leveraged towards model parsing. To test this, we train an extensive collection of classifiers, referred to here as *parsers*, to predict each of the experimental hyperparameters of our custom-trained SISR models.

In real-world applications, such parsers would be used to reverse engineer the hyperparameters of unknown SISR models, which may use architectures, loss functions, etc. from outside our parser's training set. Therefore, we evaluate our parsers on models with some hyperparameter value which we exclude from the parser's training set. We call this excluded value the *test hyperparameter value*. For example, if a parser's test hyperparameter value is RCAN, this means the parser was trained on the EDSR and RDN models, and tested on the RCAN ones.

If we have multiple super-resolved images which are known to originate from the same SISR model, then we can aggregate predictions of model hyperparameters from across multiple super-resolved images. We do this by feeding 10 randomly sampled images from the same SISR model through our parser and choosing the most popular prediction. This procedure reduces the variance in our predictions, and improves accuracy. Table 3 shows both the 1-image and 10-image accuracy of each of our model parsers.

[3] All models with seeds 2 and 3 were trained with the DIV2K dataset. Therefore, to keep the class frequencies balanced, models with seeds 2 and 3 were excluded during both training and testing of the dataset parsers.

Table 3: single-image/10-image test accuracy (%) of 19 parameter classifiers. The "chance baseline" column shows the percent chance of predicting the parameter correctly by random guess. We do not train any parsers to predict the test hyperparameter value, hence the dashes.

Predicted hyperparam.	Chance baseline	Test hyperparameter value				
		L_1	RCAN	ResNet+adv.	Flickr2K	s3
scale	50.0	46.8/50.0	99.5/100.0	96.6/100.0	99.4/100.0	99.6/100.0
loss	33.3	–	90.4/98.8	–	92.3/99.8	86.9/95.9
arch.	33.3	31.8/14.7	–	53.0/66.6	58.3/80.8	53.8/69.2
dataset [3]	25.0	28.5/28.6	36.9/51.8	29.0/28.5	–	–

The L_1 column of Table 3 shows the test accuracies of parsers which were trained on adversarially trained models (VGG+adv. and ResNet+adv.) and tested on L_1-optimized models. It appears that L_1-optimized models are too different from adversarially trained models for the parser to generalize: accuracy is no better than chance. Aggregating predictions from across multiple images offers no improvement. With aggregation, the architecture parser actually does worse, due to a bias that becomes more pronounced when aggregation reduces the variance.

Generalization from one adversarial loss function to another works better. The ResNet+adv. column shows the test accuracies of parsers which were trained on models with L_1 or VGG+adv. loss, and tested on ResNet+adv. models. Both the scale and architecture parsers do significantly better than chance. We hypothesize that this is because VGG loss and ResNet loss are not so different: both are perceptual losses trained from the ImageNet dataset. Parser generalization for the other test hyperparameter values (RCAN, Flickr2K, and s3) appears comparably good.

Below, we briefly discuss our results for each parameter that we attempt to predict. These sections are ordered from easiest to predict (scale) to the hardest (dataset).

Scale: Excluding the L_1 test hyperparameter value, our scale parsers are highly successful, obtaining accuracies between 96.6% and 99.6% depending on which parameter is being generalized over.

Loss: Our three loss functions, (L_1, ResNet+adv., and VGG+adv.) can also be distinguished with high accuracy. L_1 loss can be distinguished from the adversarial losses with near 100% accuracy, while the adversarial losses can be distinguished from each other with 84.6% accuracy.

Architecture: We had less success identifying which architecture (EDSR, RDN, or RCAN) was used to generate a given super-resolution image. Averaging across the ResNet+adv., Flickr2K, and s3 test splits, the average architecture classification accuracy is only 54.5% (against a chance accuracy of 33%). Interestingly, this classification problem was no easier for adversarially trained models than L_1-optimized ones, but it was easier for 4X scale SISR models (60.0% accuracy) than 2X scale ones (49.1%). This suggests that the additional information in an adversarial fingerprint is uncorrelated with the model architecture.

Dataset: Dataset prediction appears even more difficult than architecture prediction: For the RCAN and ResNet+adv. test hyperparameter values, we can only predict the dataset with 29.0% accuracy (against a chance accuracy of 25%). Again, this doesn't vary significantly based on the loss function used, but is significantly easier for 4X (39.0%) than for 2X (19.0%). We included the $\frac{1}{4}$ DIV2K and $\frac{1}{4}$ Flickr2K training sets in our SISR model dataset to test the hypothesis that smaller datasets would lead to more idiosyncratic SISR networks that would be easier to distinguish. However, we find that the $\frac{1}{4}$-datasets are no easier to identify (27.9% accurate) than the full datasets (30.1% accurate). In fact, the quality of quarter- and full-dataset SISR models hardly differs: On average, a full-dataset trained SISR model has a PSNR just 0.21 points higher than its $\frac{1}{4}$-dataset counterpart, and an LPIPs score just 0.0007 points lower.

4.2.1 PRETRAINED MODEL PARSING

To study model parsing for pretrained models, we train parsers on all 108 custom-trained SISR methods to predict the models' scale, loss, and architecture. (We omit the dataset parser because DIV2K is the only dataset shared by both the custom and pretrained models) We then apply these parsers to images from the 16 pretrained SISR models.

Figure 4 presents a full table of the model parser predictions for each pretrained SISR model. As with the custom model parsers, we find it easy to parse the scale of models with loss functions in the training set. The scale parser we used here was trained on L_1, VGG+adv. and ResNet+adv. losses. This parser can predict the scale of our L_1-optimized pretrained models with 100% accuracy. EnhanceNet and ProSR are also easy to parse, and are trained with losses very similar to VGG+adv. loss: both leverage a pretrained VGG network for perceptual loss, in combination with adversarial loss. But as shown in Table 3 with test hyperparameter value L_1, prediction accuracy for unseen loss functions can be much worse. NCSR and SPSR use very different approaches to super-resolution than the models in the parser's training set, and generalization to these models is accordingly worse. ESRGAN also uses a form of VGG loss, but our scale parser still performs poorly on it. ESRGAN does introduce some significant changes VGG+adv. loss, and perhaps this explains the failure to generalize, but we consider this to be an open question.

Our loss classifier can easily distinguish between L_1-optimized and adversarially trained models: all L_1-optimized pretrained models are identified as such with 100% accuracy. The true loss functions of the adversarial SISR models (ESRGAN, EnhanceNet, ProSRGAN, NCSR, and SPSR) are not in the loss classifier's training set. Yet the loss classifier always predicts that these methods were produced with an adversarial loss function (either ResNet+adv. or VGG+adv.).

Architecture prediction is unsuccessful. Our set of pretrained models contains six models whose architecture was in the training distribution: EDSR, RDN, and RCAN at 2x and 4x scale. Among these models, the architecture classifier can predict the architecture correctly just 41.0% of the time, barely exceeding the random chance classification accuracy of 33.3%.

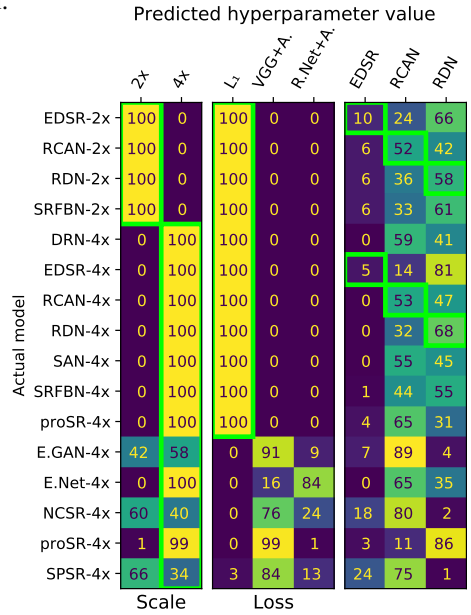


Figure 4: Parser predictions for the pretrained models. For example, out of the 100 test images for SPSR, 66 were predicted to come from a model with 2x scale, 34 from 4x. Actual model hyperparameter values are in green boxes. Some models use losses and architectures which aren't in the parsers' set of class labels, so none of their columns are inscribed in green.

4.3 Key Findings

4x SISR models produce more distinctive fingerprints than 2x for both model attribution and parsing. As has been shown for GANs, any small change to a SISR model's hyperparameters is sufficient to detect a unique fingerprint. These fingerprints are considerably more distinctive for adversarially trained models than L_1 models. However, the uniquely identifying information in an adversarially trained model's fingerprint is uncorrelated with its hyperparameters: model parsing is no easier for for adversarially trained models than L_1 ones. Using our simple classification method, it is possible to parse a SISR model's scale and loss function in a reliable and generalizable way. We achieved mixed, inconclusive results for parsing the model architecture, and were unable to effectively parse the model dataset.

5 Conclusion

We have presented the first exploration of model fingerprinting and image attribution specifically focused on single image super-resolution networks. We create a dataset of 124 super-resolution methods. We show that the extra information provided by higher upscaling factors is strongly correlated with the tractability of these problems. We show that, similarly to GANs, the fingerprints of adversarially trained models are highly sensitive to small changes in the training procedure. Our attribution classifiers learn to detect distinctive fingerprints even for SISR models outside the training set. Our model parsing experiments show promising results for parsing the loss function of the SISR model, and less promise for other parameters. We have demonstrated that these results mostly extend from our custom-trained SISR models to a set of 16 real-world pretrained SISR models.

REFERENCES

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018.

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017.

Michael Albright and S. McCloskey. Source generator attribution via inversion. *CVPR Workshops*, 2019.

Vishal Asnani, Xi Yin, Tal Hassner, and Xiaoming Liu. Reverse engineering of generative models: Inferring model hyperparameters from generated images. *arXiv/2106.07873*, 2021.

Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv/1910.08854*, 2019.

Michael Goebel, L. Nataraj, Tejaswi Nanjundaswamy, T. Mohammed, S. Chandrasekaran, and B. S. Manjunath. Detection, attribution and localization of gan generated images. *arXiv/2007.10466*, 2020.

Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Fighting deepfake by exposing the convolutional traces on images. *IEEE Access*, 2020.

Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

J. Hayes, Krishnamurthy Dvijotham, Yutian Chen, S. Dieleman, P. Kohli, and Norman Casagrande. Towards transformation-resilient provenance detection of digital media. *arXiv/2011.07355*, 2020.

Younggeun Kim and Donghee Son. Noise conditional flow model for learning the super-resolution space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017.

Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *IEEE Conference on Multimedia Information Processing and Retrieval*, 2019.

Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

J.L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *15th International Conference on Pattern Recognition*, 2000.

Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *IEEE International Conference on Computer Vision*, 2019.

Mehdi S. M. Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *IEEE International Conference on Computer Vision*, 2017.

Vladislav Skripniuk, Ning Yu, Sahar Abdelnabi, and Mario Fritz. Black-box watermarking for generative adversarial networks. *ArXiv/2007.08457*, 2020.

Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 2020.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops*, September 2018a.

Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, O. Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018b.

Xinsheng Xuan, B. Peng, Wei Wang, and Jing Dong. Scalable fine-grained generated image classification based on deep metric learning. *ArXiv/1912.11082*, 2019.

Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *IEEE International Conference on Computer Vision*, 2019.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial gan fingerprints: Rooting deepfake attribution in training data. *arXiv*, 2020.

Baiwu Zhang, Jin Peng Zhou, Ilia Shumailov, and Nicolas Papernot. On attribution of deepfakes, 2021.

J. Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *AAAI Conference on Artificial Intelligence*, 2020.

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, 2018a.

Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018b.

A Appendix

A.1 Pretrained SISR models

Our 16 pretrained SISR models come from deep-learning based SISR research papers who published their models, along with pretrained weight files, on Github. We only include methods which solve the most standard form of the single image super-resolution problem: inverting a bicubic downsampling function. Other forms of super-resolution, such as blind super-resolution or real-world super-resolution, are not included. Table 4 shows a full list of the pretrained models we use in this paper.

Table 4: 12 papers which provide the 16 pretrained super-resolution models we use in our dataset (some papers provide both 2x and 4x models).

Publication	Name	Loss	Training Dataset	Scale(s)
Lim et al. (2017)	EDSR	L_1	DIV2K	2x, 4x
Sajjadi et al. (2017)	EnhanceNet	Adv.	MSCOCO	4x
Zhang et al. (2018b)	RDN	L_1	DIV2K	2x, 4x
Wang et al. (2018b)	ProSR	L_1	DIV2K	4x
Wang et al. (2018b)	ProGanSR	Adv.	DIV2K	4x
Zhang et al. (2018a)	RCAN	L_1	DIV2K	2x, 4x
Wang et al. (2018a)	ESRGAN	Adv.	DIV2K, Flickr2K, OST	4x
Dai et al. (2019)	SAN	L_1	DIV2K	4x
Li et al. (2019)	SRFBN	L_1	DIV2K	2x, 4x
Ma et al. (2020)	SPSR	Adv.	DIV2K	4x
Guo et al. (2020)	DRN	L_1	DIV2K	4x
Kim & Son (2021)	NCSR	Adv.	DIV2K	4x

A.2 Comparison of Flickr1K, Flickr2K, and DIV2K datasets

In the DIV2K introductory paper, Agustsson & Timofte (2017) provide a few summary statistics for their dataset. Here in Table 5, we provide the same statistical analysis of our Flickr1K dataset, and the Flickr2K dataset from Lim et al. (2017). Notice that the metrics are largely similar across datasets, with the exception of pixels per image (ppi), because our images are approximately one quarter the size of those in the Flickr2K and DIV2K datasets.

Table 5: Main Characteristics of SR Datsets: number of images, pixels per image (ppi), bits per pixel using PNG compression (bpp PNG), and shannon-entropy of the images' greyscale histograms (entropy). for We report average ($\pm$standard deviation).

Dataset	Images	Ppi	Bpp PNG	Entropy
Flickr2K train	800	2793045	$12.71(\pm 2.53)$	$7.34(\pm 0.57)$
Flickr2K test	100	2794881	$12.52(\pm 2.45)$	$7.38\ (\pm 0.53)$
Flickr2K validation	100	2749737	$12.96(\pm 2.67)$	$7.52(\pm 0.29)$
DIV2K train	800	2779971	$12.68(\pm 2.79)$	$7.48(\pm 0.34)$
DIV2K validation	100	2733370	$13.24\ (\pm 2.87)$	$7.51(\pm 0.43)$
Flickr1K train	100	649958	$12.93\ (\pm 2.75)$	$7.30(\pm 0.45)$
Flickr1K test	798	646188	$12.89(\pm 2.44)$	$7.32(\pm 0.52)$
Flickr1K validation	100	633369	$12.39\ (\pm 2.39)$	$7.24(\pm 0.54)$

A.3 ResNet-Based Perceptual Loss

Our ResNet-based perceptual loss is computed by passing both the super-resolved image and the ground-truth high-resolution image through an instance of the ResNet classifier pretrained for ImageNet classificaiton. We extract the feature vectors from the last layer before classification, and take the L2 distance between HR and SR feature vectors as our ResNet loss term.

A.4 Comparison of Pretrained vs. Custom-trained Super-Resolution Quality

Judging by PSNR and LPIPS, our custom-trained L_1-optimized SISR models appear to be about on par with the pretrained models, perhaps slightly lower quality. Table 6 compares pretrained EDSR, RDN, and RCAN models to their custom-trained counterparts.

A.5 Model Parsing Using KNNs of Acutance

How much of the information we extract from our model parsers is purely a function of the blurriness/sharpness of the image? We can tell from Table 4 that 2x models tend to produce sharper

Table 6: Comparison of pretrained vs. custom-trained model performance by PSNR (higher is better) and LPIPS (lower is better) All models are 4x scale. Rows of the form {architecture}/{loss} are averaged across all custom-trained models with that combination of architecture and loss.

model	psnr	lpips
EDSR (pretrained)	29.2	0.282
EDSR (custom)	29.1	0.284
RDN (pretrained)	29.35	0.280
RDN (custom)	29.15	0.280
RCAN (pretrained)	29.4	0.270
RCAN (custom)	29.1	0.29
ESRGAN (pretrained)	28.5	0.147
EnhanceNet (pretrained)	27.5	0.196
EDSR/VGG+GAN (custom)	27.2	0.178
RDN/VGG+GAN (custom)	28.1	0.171
RCAN/VGG+GAN (custom)	28.36	0.160
EDSR/ResNet+GAN (custom)	27.23	0.180
RDN/ResNet+GAN (custom)	26.7	0.180
RCAN/ResNet+GAN (custom)	27.9	0.171

Table 7: Accuracy (%) of the parameter classification achieved using acutance as the only feature, and k-nearest neighbors as the classification scheme.

parameter	chance	L_1	RCAN	ResNet GAN	Flickr2K	s3
arch.	33.3	34.4	–	35.6	33.3	33.6
dataset	25.0	25.6	24.5	25.3	–	–
scale	50.0	48.5	58.4	50.8	59.6	60.5
loss	33.3	–	46.0	–	47.1	48.4

images than 4x, and adversarially trained models produce sharper images than L1 models. We define the acutance, or sharpness, of an image, as the variance of the laplacian of the greyscale version of that image. This simple yet robuts acutance metric originated with Pech-Pacheco et al. (2000). For each model parsing classification task we train an XceptionNet for, we also create an "acutance-based classifier" to compare it to. This baseline classifier uses the k-nearest neighbors classification algorithm (with k=20) to predict the target parameter based off of a single input feature: image *acutance*. The KNN-classifiers predict the class of each test image by finding the 20 images in the training set with the closest acutance and selecting the modal class from among those 20 images as its prediction. Table 7 shows how this classificaiton scheme performs for each of our model parsing problems.

A.6 Alternative Classifier Architectures

The parameter classifiers are clearly overfit to the training SISR models. The classifier trained to predict the architecture of the models with seeds 1 and 2 can correctly predict the architecture with 94.9% accuracy for models in its training set. Accuracy drops precipitously for the testing set: There, model architecture can only be correctly predicted with 53.8% accuracy. In an attempt to alleviate this overfitting issue, we tried training several smaller classifiers on this problem. Table 9 demonstrates the result of this experiment: no matter the network size, this trend remains the same.

We experimented with a handful of pretrained classifier backbone architectures for our model attribution and parsing networks. Table 8 shows the accuracy of several backbone architectures we trained to perform model attribution between 36 different models: the slice of models where the dataset is DIV2K and the seed isn't S3.

To make smaller versions of the XceptionNetwork, we reduce the network's width and depth. Table 9 shows four versions of XceptionNet, all trained to classify network architecture. The maximum feature width and maximum number of blocks are varied.

Table 8: Accuracies of different classifier backbones.

Classifier	Accuracy (%)
Xception	95
mobilenet	92
resnet18	79
resnet50	84

Table 9: Size and shape of the smaller versions of XceptionNet.

Version	# blocks	Max. conv. depth	# parameters	train acc (%)	test acc (%)
normal	12	728	21M	85	47
small	8	728	14.4M	86	40
smaller	8	364	3.9M	88	44
smallest	5	182	0.9M	81	41