The Best of Both Worlds: A Framework for Combining Degradation Prediction with High Performance Super-Resolution Networks

Matthew Aquilina 1,2,*, Keith George Ciantar 1,3, Christian Galea 1, Reuben A. Farrugia 1 and John Abela 5

1 Department of Communications & Computer Engineering, Faculty of ICT, University of Malta, MSD2080 Msida, Malta
2 Deanery of Molecular, Genetic & Population Health Sciences, University of Edinburgh, Edinburgh EH9 3DW, UK
3 Ascent, 90/3, Alpha Centre, Tarxien Road, LQA1815 Luqa, Malta
4 Department of Systems & Control Engineering, Faculty of Engineering, University of Malta, MSD2080 Msida, Malta
5 Department of Computer Information Systems, Faculty of ICT, University of Malta, MSD2080 Msida, Malta

* Correspondence: matthew.aquilina@um.edu.mt or m.aquilina@ed.ac.uk

Abstract: To date, the best-performing blind super-resolution (SR) techniques follow one of two paradigms: (A) train standard SR networks on synthetic low-resolution–high-resolution (LR–HR) pairs or (B) predict the degradations of an LR image and then use these to inform a customised SR network. Despite significant progress, subscribers to the former miss out on useful degradation information and followers of the latter rely on weaker SR networks, which are significantly outperformed by the latest architectural advancements. In this work, we present a framework for combining any blind SR prediction mechanism with any deep SR network. We show that a single lightweight metadata insertion block together with a degradation prediction mechanism can allow non-blind SR architectures to rival or outperform state-of-the-art dedicated blind SR networks. We implement various contrastive and iterative degradation prediction schemes and show they are readily compatible with high-performance SR networks such as RCAN and HAN within our framework. Furthermore, we demonstrate our framework’s robustness by successfully performing blind SR on images degraded with blurring, noise and compression. This represents the first explicit combined blind prediction and SR of images degraded with such a complex pipeline, acting as a baseline for further advancements.

Keywords: blind super-resolution; meta-attention; degradation prediction; metadata fusion; iterative prediction; contrastive learning; deep learning

1. Introduction

Super-Resolution (SR) is the process by which a Low-Resolution (LR) image is up-scaled, with the aim of enhancing both the image’s quality and level of detail. This operation enables the exposure of previously-hidden information which can then subsequently be used to improve the performance of any tasks depending on the super-resolved image. SR is thus highly desirable in a vast number of important applications such as medical imaging [1,2], remote sensing [3–5], and in the identification of criminals depicted in Closed-Circuit Television (CCTV) cameras during forensic investigations [6,7].

Single Image SR (SISR) is typically formulated as the restoration of High-Resolution (HR) images that have been bicubically downsampled or blurred and downsampled. On these types of LR images, state-of-the-art (SOTA) SR models can achieve extremely high performance, either by optimising for high pixel fidelity to the HR image [8–13], or by improving perceptual quality [14–16]. However, real-world images are often affected by...
additional factors such as sensor noise, complex blurring, and compression [17–19], which further deteriorate the image content and make the restoration process significantly more difficult. Moreover, many SR methods are trained on synthetically generated pairwise LR–HR images which only model a subset of the potential degradations encountered in real-world imaging systems [18,20]. As a result, the domain gap between synthetic and realistic data often causes such SR methods to perform poorly in the real world, hindering their practical use [6,18,21].

The field of blind SR is actively attempting to design techniques for image restoration which can deal with more realistic images containing unknown and complex degradations [18]. These methods often break down the problem by first estimating the degradations within an image, after which this prediction is used to improve the performance of an associated SR model. Prediction systems can range from the explicit, such as estimating the shape/size of a specific blur kernel, to the implicit, such as the abstract representation of a degradation within a Deep Neural Network (DNN) [18]. In the explicit domain, significant progress has been made in improving the accuracy and reliability of the degradation parameter estimation process. Recent mechanisms based on iterative improvement [22,23] and contrastive learning [24,25] have been capable of predicting the shape, size and noise of applied blur kernels with little to no error. However, such methods then go on to apply their prediction mechanisms with SR architectures that are smaller and less sophisticated than those used for SOTA non-blind SR.

In this work, we investigate how blind degradation prediction systems can be combined with any SR network with a Convolutional Neural Network (CNN) component, regardless of the network architecture. A robust system for the integration of the blind and SR components would allow for new techniques (both SR architectures or prediction mechanisms) to be immediately integrated and assessed under a blind setting. This would expedite and standardise the blind SR evaluation process, as well as allow any new mechanism to benefit from the latest SOTA architectures without requiring a complete redesign.

In our approach, we use a metadata insertion block to link the prediction and SR mechanisms, an operation which interfaces degradation vectors with SR network feature maps. We implement a variety of SR architectures and integrate these with the latest techniques for contrastive learning and iterative degradation prediction. Our results show that by using just a single Meta-Attention (MA) layer [26], high-performance SR models such as the Residual Channel Attention Network (RCAN) [8] and the Holistic Attention Network (HAN) [10] can be infused with degradation information to yield SR results which outperform those of the original blind SR networks trained under the same conditions.

We further extend our premise by performing blind degradation prediction and SR on images with blurring, noise and compression, constituting a significantly more complex degradation pipeline than that studied to-date by other prediction networks [22–25,27]. We show that, even on such a difficult dataset, our framework is still capable of generating improved SR performance when combined with a suitable degradation prediction system.

The main contributions of this paper are thus as follows:

1. A framework for the integration of degradation prediction systems into SOTA non-blind SR networks.
2. A comprehensive evaluation of different methods for the insertion of blur kernel metadata into CNN SR networks. Specifically, our results show that simple metadata insertion blocks (such as MA) can match the performance of more complex metadata insertion systems when used in conjunction with large SR networks.
3. Blind SR results using a combination of non-blind SR networks and SOTA degradation prediction systems. These hybrid models show improved performance over both the original SR network and the original blind prediction system.
4. A thorough comparison of unsupervised, semi-supervised and supervised degradation prediction methods for both simple and complex degradation pipelines.
5. The successful application of combined (i) blind degradation prediction and (ii) SR of images, degraded with a complex pipeline involving multiple types of noise, blurring
and compression that is more reflective of real-world applications than considered by most SISR approaches.

The rest of this paper is organised as follows: Section 2 provides an overview of related work on general and blind SR, including the methods selected for our framework. Section 3 follows up with a detailed description of our proposed methodology for combining degradation prediction methods with SOTA SISR architectures. Our framework implementation details, evaluation protocol, degradation prediction and SR results are presented and discussed in Section 4. Finally, Section 5 provides concluding remarks and potential areas for further exploration.

2. Related Work

A vast number of methods have been proposed for SR, from the seminal Super-Resolution Convolutional Neural Network (SRCNN) [28] to more advanced networks such as RCAN [8], HAN [10], Second-Order Attention Network (SAN) [9], Super-Resolution GAN (SRGAN) [14], and Enhanced SRGAN (ESRGAN) [15] among others. Domain-specific methods have also been implemented, such as those geared for the super-resolution of face images (including Super-Face Alignment Network (Super-FAN) [29] and the methods proposed in [7,30–36]), and satellite imagery [3,4,37] among others.

Most approaches derive an LR image from a HR image using a degradation model, which formulates how this process is performed and the relationship between the LR and HR images. Hence, an overview of common degradation models, including the one used as the basis for the proposed SR framework, is first provided and discussed. Given that the proposed framework combines techniques designed for both general-purpose SR and blind SR, an overview of popular and SOTA networks for both methodologies will then be provided.

2.1. Degradation Models

Numerous works in the literature have focused on the degradation models considered in the SR process, which define how HR images are degraded to yield LR images. However, the formation of an LR image \( I_{LR} \) can be generally expressed by the application of a function \( f \) on the HR image \( I_{HR} \), as follows:

\[
I_{LR} = f(I_{HR}, \theta_D)
\]  

where \( \theta_D \) is the set of degradation parameters which, in practice, are unknown.

The function \( f \) can be expanded to consider the general set of degradations applied to \( I_{HR} \), yielding the ‘classical’ degradation model as follows [6,18–20,22,23,38–43]:

\[
I_{LR} = (I_{HR} \odot k) \downarrow_s + n
\]  

where \( \odot \) represents the convolution operation, \( k \) is a kernel (typically a Gaussian blurring kernel, although it can also represent other functions such as the Point Spread Function (PSF)), \( n \) represents additive noise, and \( \downarrow_s \) is a downsampling operation (typically assumed to be bicubic downsampling [18]) with scale factor \( s \). However, this model has been criticised for being too simplistic and unable to generalise well to more complex degradations that are found in real-world images, thereby causing substantial performance losses when SR methods based on this degradation model are applied to non-synthetic images [17,19,39]. More complex and realistic degradation types have thus been considered, such as compression (which is typically signal-dependent and non-uniform, in contrast to the other degradations considered [19]), to yield a more general degradation model [17,18,44]:

\[
I_{LR} = ((I_{HR} \odot k) \downarrow_s + n)C
\]

where \( C \) is a compression scheme such as JPEG.
The aim of SR is then to approximate the inverse function of \( f \), denoted by \( f^{-1} \). This function can be applied on the LR image \( I^{LR} \) to reverse the degradation process and yield an image \( \hat{I}^{HR} \) approximating the original image \( I^{HR} \):

\[
\hat{I}^{HR} = f^{-1}(I^{LR}, \theta_R) \approx I^{HR}
\]

(4) where \( \theta_R \) represents the parameter set defining the reconstruction process. This degradation model forms the basis of the proposed SR framework.

Other works have extended the general model in Equation (3) to more complex cases. One such work synthesizes training pairs using a ‘high-order’ degradation process where the degradation model is applied more than once [16]. The authors of [19] proposed a ‘practical’ degradation model to train the ESRGAN-based BSRNet and BSRGAN models [19], which consider multiple Gaussian blur kernels, downsampling operators, noise levels modelled by Additive White Gaussian Noise (AWGN), processed camera sensor noise types, and quality factors of JPEG compression. Random shuffling of the order in which the degradations are applied is also performed.

Counter-arguments to these complex models have also been made. For instance, the authors of [17] argued that ‘practical’ degradation models as proposed in [16,19] (so called because a wide variety of degradations are considered, reflecting practical real-world applications) may achieve promising results on complex degradations but then ignore easier edge cases. Given that such cases primarily entail combinations of degradation subsets, a gated degradation model is proposed whereby the base degradations to be applied are randomly selected. Since the magnitudes of some degradations in the proposed framework may be reduced to the point where they are practically negligible, the degradation model forming the basis of this work (which is based on the processes defined in Equations (3) and (4) as previously discussed) can be said to approximate this gated mechanism. This also means that the proposed framework considers degradations found in practical real-world applications.

2.2. Non-Blind SR Methods

Most methods proposed for SR have tended to focus on the case where degradations are assumed to be known, either by designing models for specific degradations or by designing approaches that are able to use supplementary information about the degradations afflicting the image. However, this information is neither estimated nor derived from the corrupted image, limiting the use of such methods in the real world where degradations are highly variable in terms of both their type and magnitude. Despite these limitations, non-blind SR methods have served an important role in enabling more rapid development of new techniques on what is arguably a simpler case of SR. An overview of notable methods will now be given.

The seminal non-blind SR method is considered to be the SRCNN network [28], which was one of the first to use deep learning and CNNs for the task of SR. However, it only consists of three layers and requires the LR image to first be upsampled using bicubic interpolation, and is now outperformed by most modern approaches.

To facilitate the training of a large number of CNN layers, the Residual Network (ResNet) architecture proposed in [45] introduced skip connections to directly feed feature maps at any level of the network to deeper layers, a process corresponding to the identity function which deep networks find hard to learn. This counteracted the problem of vanishing gradients apparent in classical deep CNN networks, allowing the authors of [45] to expand their network size without impacting training performance. ResNet was extended to SR in [14], to create the Super-Resolution Residual Network (SRResNet) approach that was also used as the basis of a Generative Adversarial Network (GAN)-based approach termed SRGAN.

SRGAN was extended in [15] to yield ESRGAN, which included the introduction of a modified adversarial loss to determine the relative ‘realness’ of an image, and not just a decision as to whether the generated image is ‘real’ or ‘fake’. ESRGAN also introduced a
modified VGG-based perceptual loss, which uses feature maps extracted from the VGG residual blocks right before the activation layers to reduce sparsity and better supervise brightness consistency and texture recovery. ESRGAN was further extended in [16] to yield Real ESRGAN (Real-ESRGAN), where the focus was on the implementation of a ‘high-order’ degradation process that allowed the application of the degradation model more than once (in contrast to most works which only apply the model one time).

Enhanced Deep Super-Resolution (EDSR) [46] was also based on ResNet and incorporated observations noted in previous works such as SRResNet, along with other novel contributions that had a large impact on subsequent CNN-based SR models. These included the removal of batch normalisation layers to disable restriction of feature values and reduce the memory usage during training that, in turn, allowed for a greater number of layers and filters to be used.

The RCAN approach proposed in [8] is composed of ‘residual groups’ that each contain a number of ‘channel attention blocks’, along with ‘long’ and ‘short’ skip connections to enable the training of very deep CNNs. The channel attention blocks enable the assignment of different levels of importance of low-frequency information across feature map channels. The concept of attention introduced by RCAN was developed further by other methods such as SAN [9] and HAN [10], which proposed techniques such as channel-wise feature re-scaling and modelling of any inter-dependencies among channels and layers.

Recently, vision transformers applied to SR were also proposed, such as the Encoder-Decoder-based Transformer (EDT) [47], Efficient SR Transformer (ESRT) [48], and the Swin Image Restoration (SwinIR) [12] approach that is based on the Swin Transformer [49]. Approaches such as Efficient Long-Range Attention Network (ELAN) [13] and Hybrid Attention Transformer (HAT) [50], which attempt to combine CNN and transformer architectures, have also been proposed with further improvements in SR performance.

An extensive exposition of generic non-blind SR methods can be found in [51,52].

2.3. Blind SR Methods

Although numerous SR methods have been proposed, a substantial number of approaches tend to employ the classical degradation model (in Equation (2)). Besides not being quite reflective of real-world degradations (as discussed in Section 2.1), a substantial number of approaches also assume that the degradations afflicting an image are known. This is largely not the case, so that such approaches tend to exhibit noticeable performance degradation on images found ‘in-the-wild’.

As a result, Blind SR methods have been designed for better robustness when faced with such difficult and unknown degradations, making them more suitable for real-world applications. There exist several types of blind SR methods, based on the type of data used and how they are modelled [18]. An overview of the various types of approaches and representative methods will now be provided.

2.3.1. Approaches Utilising Supplementary Attributes for SR

Early work focused on the development of methods that are directly supplied with ground-truth information on degradations. The focus then turned on developing techniques to best utilise this degradation information (as opposed to non-blind SR methods which do not use any form of supplementary information). Approaches of this kind generally consider the classical degradation model [18].

Notable methods incorporating metadata information in networks include Super-Resolution network for Multiple Degradations (SRMD) [39], Unified Dynamic Convolutional Network for Variational Degradations (UDVD) [53], the Deep Plug-and-Play SR (DPSR) framework, and the approach in [54]. Each of these approaches showed that SR networks could improve their performance by using this degradation information. Frameworks enabling the extension of existing non-blind SR methods to use degradation information have also been proposed, such as the ‘meta-attention’ approach in [26].
and Conditional hyper-network framework for SR with Multiple Degradations (CMDSR) proposed in [55].

These methods clearly show the plausibility of improving SR performance with degradation metadata. However, apart from requiring some means of obtaining the degradation information, such methods are also highly reliant on the quality of the information input to the networks, which is not a trivial task. Moreover, any deviations in the estimated inputs can lead to kernel mismatches that may be detrimental to the SR performance [18,23,38].

2.3.2. Iterative Kernel Estimation Methods

Blur kernel estimation during the SR process is one of the most common blind SR prediction tasks, and alleviates the problem of kernel mismatches present in methods such as SRMD as described above. Often, iterative mechanisms are applied for direct kernel estimation. One such method is Iterative Kernel Correction (IKC) [22], which leverages the observation that kernel mismatch tends to produce regular patterns by estimating the degradation kernel and correcting it in an iterative fashion using a corrector network. In this way, an acceptable result is progressively approached. The authors of [22] also proposed a non-blind SR network, named Spatial Feature Transform Multiple Degradations (SFTMD), which was shown to outperform existing methods (such as SRMD) that insert blur kernel metadata into the SR process.

The Deep Alternating Network (DAN) [23] method (also known as DANv1) and its updated version DANv2 [38] build upon the IKC approach, by combining the SR and kernel corrector networks within a single end-to-end trainable network. The corrector was also modified to use the LR input conditioned on intermediate super-resolved images, instead of conditioning these images on the estimated kernel as done in IKC. The Kernel-Oriented Adaptive Local Adjustment network (KOALAnet) [56] is able to adapt to spatially-variant characteristics within an image, which allows a distinction to be made between blur caused by undesirable effects, and between blur introduced intentionally for aesthetic purposes (e.g., the Bokeh effect). However, given that such methods still rely on kernel estimation, they also exhibit poor performance when evaluated on images having different degradations than those used to train the model.

2.3.3. Training SR Models on a Single Image

Another group of methods such as KernelGAN [57] and Zero-Shot SR (ZSSR) [58] use intra- and inter-scale recurrence of patches, based on the internal statistics of natural images, to construct an individual model for each input LR image. Hence, the data used for training is that which is present internally within the image being super-resolved, circumventing the need to use an external dataset of images.

Such methods tend to assume that a downscaled version of a patch within a LR image should have a similar distribution to the patch in the original LR image. However, the assumption of recurring patches within and across scales may not hold true for all images (such as those containing a wide variety of content) [18].

2.3.4. Implicit Degradation Modelling

Modelling an explicit combination of multiple degradation types can be a very complex task on images found ‘in-the-wild’. Hence, some approaches have also attempted to implicitly model the degradation process by comparing the data distribution of real-world LR image sets with synthetically generated ‘clean’ datasets (containing limited or no degradations) [18]. Methods are typically based on GANs, such as Cycle-in-Cycle GAN (CinCGAN) [59] and the approaches in [60,61], and do not require a HR reference for training.

However, one of the drawbacks of this type of method is that they tend to require vast amounts of data, which may not always be available. Some approaches, such as Degradation GAN [62] and Frequency Separation for real-world SR (FSSR) [63], attempt to counteract this issue by learning the HR-to-LR degradation process to generate realistic LR
samples that can be used during the SR model training. However, most models designed for implicit degradation modelling use GANs which are known to be hard to train and may introduce fake textures or artefacts that can be detrimental for some real-world applications [18].

2.3.5. Contrastive Learning

In the image classification domain, DNNs are known to be highly capable of learning invariant representations, enabling the construction of good classifiers [64]. However, it has been argued that DNNs are actually too eager to learn invariances [64]. This is because they often learn only the features necessary to discriminate between classes but then fail to generalise well to new unseen classes when employing a supervised setting in what is known as supervision collapse [64,65]. Indeed, the ubiquitous cross-entropy loss used to train supervised deep classifier models has received criticism for several shortcomings [66], such as its sensitivity to noisy labels [67,68] and the possibility of poor margins [69–71].

Contrastive learning techniques, mostly developed in the Natural Language Processing (NLP) domain, have recently seen a resurgence and have driven significant advances in self-supervised representation learning in an attempt to mitigate these issues [66]. Contrastive learning is a self-supervised approach where models are trained by comparing and contrasting ‘positive’ image pairs with ‘negative’ pairs [72]. Positive images can be easily created by applying augmentations to a source image (e.g., flipping, rotations, colour jitter etc.). In the SR domain, positive samples are typically patches extracted from within the same image, while crops taken from other images are labelled as negative examples [24,72,73].

One such contrastive learning-based approach is Momentum Contrast (MoCo), proposed in [73] for the tasks of object detection, classification, and segmentation. MoCo employs a large queue of data samples to enable the use of a dictionary (containing samples observed in preceding mini-batches) which is much larger than the mini-batch size. However, since a large dictionary also makes it intractable to update the network parameters using back-propagation, a momentum update which tightly controls the parameters’ rate of change is also proposed. MoCo was extended in [74] to yield MoCov2, based on design improvements proposed for SimCLR [72]. The two main modifications constitute the replacement of the fully-connected layer at the head of the network with an Multi-Layer Perceptron (MLP) head, and the inclusion of blur augmentation.

Supervised MoCo (SupMoCo) [64] was also proposed as an extension of MoCo, whereby class labels are additionally utilised to enable intra-class variations to be learnt whilst retaining knowledge on distinctive features acquired by the self-supervised components. SupMoCo was shown to outperform the Supervised Contrastive (SupCon) approach [66], which also applied supervision to SimCLR [72].

Such self-supervised methods have seen limited use in the SR domain thus far. However, the promising performance demonstrated in other domains could encourage further research and development in the SR arena. One of the first SR networks to use contrastive learning for blind SR was the Degradation-Aware SR (DASR) network [24], where an unsupervised content-invariant degradation representation is obtained in a latent feature space where the mutual information among all samples is maximised.

In [25], Implicit Degradation Modelling Blind SR (IDMBSR), considers the degrees of difference in degradation between a query image and negative exemplars, in order to determine the amount of ‘push’ to exert. Specifically, the greater the difference between a query and a negative example, the greater the push. In this way, degradation information is used as weak supervision of the representation learning to improve the network’s ability to characterise image degradations.

2.3.6. Other Methodologies

The mechanisms discussed in this section are some of the most prominent in the blind SR literature. However, many other modalities exist which do not neatly fall into any single category. One such method is Mixture of Experts SR (MoESR) [42], where a panel of expert
predictors is used to help optimise the prediction system for blur kernels of different shapes and sizes. A comprehensive overview of the state of blind SR research can be found in [18].

2.4. Conclusions

As described above, there exist numerous algorithms, architectures, and frameworks designed to perform the task of SR, both for non-blind and blind scenarios. However, current approaches for both tasks suffer from major drawbacks. On the one hand, non-blind SR architectures have been optimised for high performance on synthetic data, but under-perform in practical degradation scenarios. On the other hand, blind SR networks and degradation prediction systems are more capable of dealing with real-world images but use weaker and more limited architectures that restrict their performance ceiling. These observations serve as the motivation to combine the best of both worlds, chiefly to leverage the performance of non-blind SR networks with the practical applications of degradation prediction systems.

Given the prominence of iterative and contrastive methods in blind SR, a number of these mechanisms were selected and implemented within the proposed framework. The principles of SupMoCo were also applied to construct a more controllable self-supervised contrastive learning function (Section 3.5.2). However, in principle, any degradation estimation system could be coupled to any SR model, using the framework described in the rest of this paper.

3. Methodology

3.1. Degradation Model

For all our analyses, we follow the degradation model described in Equation (3) that encompasses some of the most common degradations found in the real world, namely blurring, downsampling, instrument noise, and compression. The task of any SR algorithm is to then essentially reverse the degradation process (as shown in Equation (4)) afflicting an LR image $I_{LR}$ to yield an image $\hat{I}_{HR}$ that approximates the original HR image ($I_{HR}$).

For the LR–HR pairs used for training and testing, a variety of different operations across each type of degradation are applied, with the full details provided in Section 4.1. Given that the order, type, and the parameters of the degradations are known in advance, the task of our degradation prediction models is significantly easier than the fully blind case with completely unknown degradations. However, the degradation prediction principles of each model could easily be extended to more complex and realistic degradation pipelines [16,19] through adjustments to the degradation modelling process. That said, a variety of degradation magnitudes emulating the ‘gated’ degradation pipeline as proposed in [17] is considered, which is more reflective of real-world scenarios as discussed in Section 2.1. Indeed, it is shown in Section 4.7 that our models are still capable of dealing with test images degraded in the real world despite the easier conditions in the training datasets.

3.2. Framework for Combining SR Models with a Blind Predictor

Our proposed general framework for combining blind SR prediction mechanisms with non-blind SR models aims to amplify the strengths of both techniques with minimal architectural changes on either side. In most cases, explicit blind prediction systems generate vectors to describe the degradations present in an image. On the other hand, the vast majority of SR networks feature convolutional layers and work in image space rather than in a vector space. To combine the two mechanisms, the prediction model was kept separate from the SR core and the two were bridged using a metadata insertion block, as shown in Figure 1. This makes it relatively simple for different prediction or SR blocks to be swapped in and out, while keeping the overall framework unchanged. We considered a variety of options for each of the three main components of the framework. The metadata insertion and degradation prediction mechanisms selected and adjusted for our framework
are discussed in the remaining sections of the methodology, while the chosen SR core networks are provided in Section 4.1.

Figure 1. Proposed framework for combining blind degradation systems and SR models. The metadata insertion block acts as the bridge between the two systems, allowing the SR model to exploit the predicted degradation to improve its performance. Depending on the blind predictor mechanism chosen, the SR image can be fed back into the predictor to help improve its accuracy.

3.3. Metadata Insertion Block

The metadata insertion block plays an essential role in our framework, since it converts a degradation vector into a format compatible with CNN-based SR networks, and ensures that this information is fully utilised throughout the SR process. Despite its importance, the inner workings of the mechanism are poorly understood since CNN-based models (SR or otherwise) are notoriously difficult to interpret. In fact, multiple methods for combining vectors and images within CNNs have been proposed and these vary significantly in complexity and size without a clear winner being evident. We selected and tested some of the most effective mechanisms in the literature within our framework, of which the following is a brief description of each (with a graphical depiction provided in Figure 2):

- **SRMD-style [39] (Figure 2A):** An early adopter of metadata insertion in SR, the SRMD technique involves the transformation of vectors as additional pseudo-image channels. Each element of the input degradation vector is expanded (by repeated tiling) into a 2D array, with the same dimensions as the input LR image. These pseudo-channels are then fed into the network along with the real image data, ensuring that all convolutional filters in the first layer have access to the degradation information. Other variants of this method, which include directly combining the pseudo-channels with CNN feature maps, have also been proposed [23]. The original SRMD network used this method for Principal Component Analysis (PCA)-reduced blur kernels and noise values. In our work, we extended this methodology to all degradation vectors considered.
• **MA** [26] (Figure 2B): MA is a trainable channel attention block which was proposed as a way to upgrade any CNN-based SR network with metadata information. Its functionality is simple—an input vector is stretched to the same size as the number of feature maps within a target CNN network using two fully-connected layers. Each vector element is normalised to lie in the closed interval \([0, 1]\) and then applied to selectively amplify its corresponding CNN feature map. MA was previously applied for PCA-reduced blur kernels and compression quality factors only. We extended this mechanism to all degradation parameters considered, by combining them into an input vector which is then fed into the MA block. The fully-connected layer sizes were then expanded as necessary to accommodate the input vector.

• **SFT** [22] (Figure 2C): The SFT block is based on the SRMD concept but with additional layers of complexity. The input vector is also stretched into pseudo-image channels, but these are added to the feature maps within the network rather than the actual original image channels. This combination of feature maps and pseudo-channels are then fed into two separate convolutional pathways, one of which is multiplied with the original feature maps and the other added on at the end of the block. This mechanism is the largest (in terms of parameter count due to the number of convolutional layers) of those considered in this paper. As with SRMD, this method has only been applied for blur kernel and noise parameter values, and we again extended the basic concept to incorporate all degradation vectors considered.

• **Degradation-aware (DA) block** [24] (Figure 2D): The DA block was proposed in combination with a contrastive-based blind SR mechanism for predicting blur kernel degradations. It uses two parallel pathways, one of which amplifies feature maps in a manner similar to MA, while the other explicitly transforms vector metadata into a 3D kernel, which is applied on the network feature maps. This architecture is highly specialised to kernel-like degradation vectors, but could still be applicable for general degradation parameters given its dual pathways. We extended the DA block to all degradation vectors as we did with the MA system.

• **Degradation-Guided Feature Modulation Block (DGFMB)** [25] (Figure 2E): This block was conceived as part of another contrastive-based network, again intended for blur and noise degradations. The main difference here is that the network feature maps are first reduced into vectors and concatenated with the degradation metadata in this form, rather than in image space. Once the vectors are concatenated, a similar mechanism to MA is used to selectively amplify the output network feature maps. As before, we extended this mechanism to other degradation parameters by combining these into an input vector.

Many of the discussed metadata insertion mechanisms were initially introduced as repeated blocks which should be distributed across the entirety of an SR network. However, this can significantly increase the complexity (in both size and speed) of a network as well as make implementation difficult, given the variety of network architectures available. Our testing has shown that, in most cases, simply adding one metadata-insertion block at the front of the network is enough to fully exploit the degradation vector information (results in Section 4.2). Further implementation details of each block are provided in Section 4.1.
Figure 2. Metadata insertion mechanisms investigated in this paper. (A) SRMD-style metadata insertion. (B) Meta-Attention block. (C) SFT block. (D) Degradation-Aware block. (E) Degradation-Guided Feature Modulation Block. The end result of each mechanism is the trainable modulation of CNN feature maps using the information present in a provided vector. Each mechanism varies substantially in complexity, positioning and the components involved.

3.4. Degradation Prediction—Iterative Mechanism

The simplest degradation prediction system tested for our framework is the DAN iterative mechanism proposed in [23]. This network consists of two convolutional submodules—a restorer, in charge of the super-resolution step and an estimator, which predicts the blur kernel applied on an LR image (in PCA reduced form). Both modules are co-dependent; the restorer produces an SR image based on the degradation prediction while the estimator makes a degradation prediction informed by the SR image. By repeatedly...
alternating between the two modules, the results of both can be iteratively improved. Furthermore, both networks can be optimised simultaneously by back-propagating the error of the SR and degradation estimates.

The iterative mechanism is straightforward to introduce into our framework. We implemented the estimator module from DAN directly, and then coupled its output with a core SR network through a metadata insertion block (Figure 1). While the authors of DAN only considered blur kernels in their work, it should be possible to direct the estimator to predict the parameters of any specified degradation directly. We tested this hypothesis for both simple and complex degradations, the results of which are provided in Sections 4.3 and 4.5.

3.5. Degradation Prediction—Contrastive Learning

Contrastive learning is another prominent method for degradation estimation in blind SR. We considered three methods for contrastive loss calculation: one completely unsupervised and two semi-supervised techniques. These mechanisms have been illustrated in Figure 3A, and are described in further detail hereunder.

3.5.1. MoCo—Unsupervised Mechanism

Contrastive learning for blind SR was first proposed in [24], where the authors used the MoCo [73] mechanism to train convolutional encoders that are able to estimate the shape and noise content in blur kernels applied on LR images. The encoder is taught to generate closely-matched vectors for images with identical or similar degradations (e.g., equally sized blur kernels) and likewise generate disparate vectors for vastly different degradations. These encoded vectors, while not directly interpretable, can be utilised by a downstream SR model to inform the SR process. The proposed MoCo encoder training mechanism works as follows:

- Two identical encoders are instantiated. One acts as the ‘query’ encoder and the other as the ‘key’ encoder. The query encoder is updated directly via backpropagation from computed loss, while the key encoder is updated through a momentum mechanism from the query encoder.
- The encoders are each directed to generate a degradation vector from one separate square patch per LR image, as shown on the right-hand side of Figure 3A. The query encoder vector is considered as the reference for loss calculation, while the key encoder vector generated from the second patch acts as a positive sample. The training objective is to drive the query vector to become more similar to the positive sample vector, while simultaneously repelling the query vector away from encodings generated from all other LR images (negative samples).
- Negative samples are generated by placing previous key encoder vectors from distinct LR images into a queue. With both positive and negative encoded vectors available, an InfoNCE-based [75] loss function can be applied:

\[
L_{MOCO} = \frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(f_q(x_1^i) \cdot f_k(x_2^i) / \tau)}{\exp(f_q(x_1^i) \cdot f_k(x_2^i) / \tau) + \sum_{j=1}^{N_{queue}} \exp(f_q(x_1^i) \cdot q_j / \tau)}
\]

where \( f_q \) and \( f_k \) are the query and key encoders, respectively, \( x_1^i \) is the first patch from the \( i \)th image in a batch (batch size \( B \)), \( q_j \) is the \( j \)th entry of the queue of size \( N_{queue} \) and \( \tau \) is a constant temperature scalar. With this loss function, the query encoder is updated to simultaneously move its prediction closer to the positive encoding, and farther away from the negative set of encodings. This push-pull effect is depicted by the coloured dotted lines within the MoCo box in Figure 3A. This loss should enable the encoder to distinguish between the different degradations present in the positive and negative samples.
Figure 3. Contrastive learning for SR. (A) The three contrastive learning mechanisms considered in this work. While MoCo is the simplest and the only fully unsupervised method, SupMoCo and WeakCon provide more targeted learning at the expense of requiring user-defined labelling systems. (B) Direct parameter regression can also be used to add an additional supervised element to the encoder training process.
In [24], only one positive patch is used per input image. However, this can be easily extended (generalised) to multiple positive patches through the following modifications to the loss function (shown in blue):

\[
L_{\text{MOCO}} = \frac{1}{B} \sum_{i=1}^{B} \left[ -\log \frac{\sum_{l=1}^{P_i} \exp \left( \frac{f_q(x^l_i) \cdot f_k(x^l_i)}{\tau} \right)}{\sum_{l=1}^{P_i} \exp \left( \frac{f_q(x^l_i) \cdot f_k(x^l_i)}{\tau} \right) + \sum_{j=1}^{N_{\text{queue}}} \exp \left( \frac{f_q(x^l_i) \cdot q^j}{\tau} \right)} \right] \tag{6}
\]

where \( P_i \) is the number of positive patches for the \( i \)th image in a batch. For most of our tests, we retain just one positive patch (using the loss from Equation (5)) to match with [24], unless indicated.

3.5.2. SupMoCo—Semi-Supervised Mechanism

In the image classification domain, advances into semi-supervised contrastive learning have resulted in further performance improvements over MoCo. One such mechanism, SupMoCo [64], provides more control over the contrastive training process. In essence, all encoded vectors can be assigned a user-defined label, including the query vector. With these labels, the contrastive loss function can be directed to push the query vector towards all the key vectors sharing the same label, while repelling it away from those that have different labels, as shown by the SupMoCo column of Figure 3A. To do this, the authors of [64] added a label-based loss component to Equation (6) which also considers the positive samples in the queue (highlighted in blue):

\[
L_{\text{SUPMOCO}} = \frac{1}{B} \sum_{i=1}^{B} \left[ -\log \frac{\sum_{l=1}^{P_i} \exp \left( \frac{f_q(x^l_i) \cdot f_k(x^l_i)}{\tau} \right) + \sum_{m=1}^{Q_i} \exp \left( \frac{f_q(x^l_i) \cdot f_k(x^m_i)}{\tau} \right)}{\sum_{l=1}^{P_i} \exp \left( \frac{f_q(x^l_i) \cdot f_k(x^l_i)}{\tau} \right) + \sum_{j=1}^{N_{\text{queue}}} \exp \left( \frac{f_q(x^l_i) \cdot q^j}{\tau} \right)} \right] \tag{7}
\]

where \( Q_i \) is the number of samples in the queue with the same label as the query vector and \( F_i = P_i + Q_i \).

This system allows more control of the trajectory of the contrastive loss, reducing inaccuracies while pushing the encoder to recognise patterns based on the labels provided. For our degradation pipeline, we implemented a decision tree which assigns a unique label to each possible combination of degradations. In brief, a label is assigned to each degradation based on a number of factors:

- Blur kernel type
- Blur kernel size; either low/high, which we refer to as double precision (2 clusters per parameter), or low/medium/high, which we refer to as triple precision (3 clusters per parameter), classification.
- Noise type
- Noise magnitude (either a double or triple precision classification)
- Compression type
- Compression magnitude (either a double or triple precision classification)

An example of how this decision tree would work for labelling of compression type and magnitude is provided in Figure 4.

The decision tree labelling should push the encoder to identify the presence of different degradation classes more quickly than in the unsupervised case. Aside from the labelling system, SupMoCo is trained in an identical fashion to MoCo, including the usage of momentum to update the key encoder. A full description of all degradations applied in our pipeline is provided in Section 4.1.
3.5.3. WeakCon—Semi-Supervised Mechanism

Another semi-supervised contrastive paradigm (which we refer to as WeakCon) has been proposed in [25]. Instead of assigning discrete labels to each degradation, the authors propose a system for modulating the strength of the contrastive loss. By calculating the difference in degradation magnitudes between query and negative samples, the negative contrastive push can be increased or decreased, according to how different the degradations are. This weighted push is illustrated in the WeakCon column of Figure 3A. In [25], the authors utilise the Euclidean distance between query and selected negative sample blur kernel widths and noise sigmas to calculate a weight $w$. Using this weighting (highlighted in blue), the authors updated the InfoNCE-style loss function in Equation (6) so that the contrastive loss can be weakly-supervised as follows:

$$L_{\text{WEAKCON}} = \frac{1}{B} \sum_{j=1}^{B} \left[ -\log \frac{\sum_{i=1}^{P_l} \exp \left( f_q(x_i^j) \cdot f_k(x_i^j) / \tau \right)}{\sum_{i=1}^{P_l} \exp \left( f_q(x_i^j) \cdot f_k(x_i^j) / \tau \right) + \sum_{j=1}^{N_{\text{queue}}} w_{ij} \exp \left( f_q(x_i^j) \cdot q_k / \tau \right)} \right]$$

(8)

where $w_{ij}$ indicates the distance between the degradations of negative sample $j$ and query sample $i$. Since the original work in [25] focused only of isotropic Gaussian kernels and Gaussian noise, we extend this special case to other degradations by similarly calculating the Euclidean distance between degradation vectors containing blur widths in both horizontal and vertical directions, noise scales and compression qualities.

3.5.4. Direct Regression Attachment

While contrastive representations can be visualised using dimensionality reduction, it is difficult to quantify their prediction accuracy with respect to the true degradation parameters. To provide further insight into the training process, we attach a further set of fully-connected layers to the contrastive encoder, as shown in Figure 3B. These layers are set to directly transform the contrastive vector into the magnitudes of the various degradations being estimated. A regression loss (L1 loss between predicted vector and target degradation magnitudes) can also be introduced as an additional supervised element. This direct parameter prediction can be easily quantified into an estimation error, which can help track training progress. We train various models with and without these extra layers, with the details provided in Section 4.
3.6. Extensions to Degradation Prediction

In both the iterative and contrastive cases, our prediction mechanisms are centred around general degradation parameter prediction and, as such, could be extended to any degradation which can be parameterised in some form. Alternatively, degradations could be represented in vector form through the use of dimensionality reduction techniques (as is often done with blur kernels, on which PCA is applied). Dimensionality reduction can also be used as an imperfect view of the differences between contrastive vectors encoded for different degradations. We provide our analyses of the contrastive process in Sections 4.3 and 4.5.

4. Experiments & Results

4.1. Implementation Details

4.1.1. Datasets and Degradations

For analysis purposes, we created two LR degradation pipelines:

- **Simple pipeline (blurring and downsampling):** For our metadata insertion screening and blind SR comparison, we used a degradation set of just Gaussian blurring and bicubic downsampling corresponding to the ‘classical’ degradation model described in Equation (2). Apart from minimising confounding factors, this allowed us to make direct comparisons with pre-trained models provided by the authors of other blind SR networks. For all scenarios, we used only $21 \times 21$ isotropic Gaussian kernels with a random width ($\sigma$) in the range $[0.2, 3]$ (as recommended in [16]), and $\times 4$ bicubic downsampling. The $\sigma$ parameter was then normalised to the closed interval $[0, 1]$ before it was passed to the models.

- **Complex pipeline:** In our extended blind SR training schemes, we used a full degradation pipeline as specified in Equation (3), i.e., sequential blurring, downsampling, noise addition and compression. For each operation in the pipeline, a configuration was randomly (from a uniform distribution) selected from the following list:
  
  - **Blurring:** As proposed in [16], we sampled blurring from a total of 7 different kernel shapes: iso/anisotropic Gaussian, iso/anisotropic generalised Gaussian, iso/anisotropic plateau, and sinc. Kernel $\sigma$ values (both vertical and horizontal) were sampled from the range $[0.2, 3]$, kernel rotation ranged from $-\pi$ to $\pi$ (all possible rotations) and the shape parameter $\beta$ ranged from $[0.5, 8]$ for both generalised Gaussian and plateau kernels. For sinc kernels, we randomly selected the cutoff frequency from the range $[\pi/5, \pi]$. All kernels were set to a size of $21 \times 21$ and, in each instance, the blur kernel shape was randomly selected, with equal probability, from the 7 available options. For a full exposition on the selection of each type of kernel, please refer to [16].
  
  - **Downsampling:** As in the initial model screening, we again retained $\times 4$ bicubic downsampling for all LR images.
  
  - **Noise addition:** Again following [16], we injected noise using one of two different mechanisms, namely Gaussian (signal independent read noise) and Poisson (signal dependent shot noise). Additionally, the noise was either independently added to each colour channel (colour noise), or applied to each channel in an identical manner (grey noise). The Gaussian and Poisson mechanisms were randomly applied with equal probability, grey noise was selected with a probability of 0.4, and the Gaussian/Poisson sigma/scale values were randomly sampled from the ranges $[1, 30]$ and $[0.05, 3]$, respectively.
  
  - **Compression:** We increased the complexity of compression used in previous works by randomly selecting from either JPEG or JM H.264 (version 19) [76] compression at runtime. For JPEG, a quality value was randomly selected from the range $[30, 95]$ (following [16]). For JM H.264, images were compressed as single-frame YUV files where a random I-slice Quantization Parameter (QPI) was selected from the range $[20, 40]$, as discussed in [26].
To allow for a fair (and direct) comparison to other works, the training, validation and testing HR datasets we selected are identical to those used in the SR works we used as baselines or comparison points (Section 4.1.2). Thus, all our models were trained on LR images generated from HR images of DIV2K [77] (800 images) and Flickr2K [78] (2650 images). Validation and best model selection were performed on the provided DIV2K validation set (100 images).

For final results comparison, the standard SR test sets Set5 [79], Set14 [80], BSDS100 [81], Manga109 [82] and Urban100 [83] were utilised. For these test images, the parameters of each degradation were explicitly selected. The exact degradation details for each scenario are specified in all the tables and figures presented. The super-resolved images were compared with the corresponding target HR images using several metrics during testing and validation, namely Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity index (SSIM) [84] (direct pixel comparison metrics) and Learned Perceptual Image Patch Similarity (LPIPS) [85] (perceptual quality metric). In all cases, images were first converted to YCbCr, and the Y channel used to compute metrics.

The degradation pipelines and further implementation details are available in our linked PyTorch [86] codebase (https://github.com/um-dsrg/RUMpy).

4.1.2. Model Implementation, Training and Validation

Due to the diversity of the models investigated in this work, a number of different training and validation schemes were followed depending on the task and network being investigated:

- **Non-blind SR model training**: For non-blind model training, we initialised the networks with the hyperparameters as recommended by their authors, unless otherwise specified. All models were trained from scratch on LR–HR pairs generated from the DIV2K and Flickr2K datasets using either the simple or complex pipelines. For the simple pipeline, one LR image was generated from each HR image. For the complex pipeline, five LR images were generated per HR image to improve the diversity of degradations available. In both cases, the LR image set was generated once and used to train all models. All simple pipeline networks were trained for 1000 epochs, whereas the complex pipeline networks were trained for 200 epochs to ensure fair comparisons (since each epoch contains 5 times as many samples as in the simple case). This training duration (in epochs) was chosen as a compromise between obtaining meaningful results and keeping the total training time low.

  For both pipelines, training was carried out on 64 × 64 LR patches. The Adam [87] optimiser was used. Variations in batch size and learning rate scheduling were made for specific models as necessary in order to ensure training stability and limit Graphical Processing Unit (GPU) memory requirements. The configurations for the non-blind SR models tested are as follows:

  - RCAN [8] and HAN [10]: We selected RCAN to act as our baseline model as a compromise between SR performance and architectural simplicity. To push performance boundaries further, we also trained and tested HAN as a representative SOTA pixel-quality CNN-based SR network. For these models, the batch size was set to 8 in most cases, and a cosine annealing scheduler [88] was used with a warm restart after every 125,000 iterations and an initial learning rate of $1 \times 10^{-4}$. Training was driven solely by the L1 loss function which compares the SR image with the target HR image. After training, the epoch checkpoint with the highest validation PSNR was selected for final testing.

  - Real-ESRGAN [16]: We selected Real-ESRGAN as a representative SOTA perceptual quality SR model. The same scheme described for the original implementation was used to train this network. This involved two phases: (i) a pre-training stage where the generator was trained with just an L1 loss, and (ii) a multi-loss stage where a discriminator and VGG perceptual loss network were introduced (further details are provided in [16]). We pre-trained the model for 715 and
150 epochs (which match the pretrain:GAN ratio as originally proposed in [16]) for the simple and complex pipelines, respectively. In both cases, the pre-training optimiser learning rate was fixed at $2 \times 10^{-4}$, while the multi-loss stage involved a fixed learning rate of $1 \times 10^{-4}$. A batch size of 8 was used in all cases. After training, the model checkpoint with the lowest validation LPIPS score in the last 10% of epochs was selected for testing.

- ELAN [13]: We also conducted a number of experiments with ELAN, a SOTA transformer-based model. For this network a batch size of 8 and a constant learning rate of $2 \times 10^{-4}$ were used in all cases. As with RCAN and HAN, the L1 loss was used to drive training and the epoch checkpoint with the highest validation PSNR was selected for final testing.

- **Iterative Blind SR**: Since the DAN iterative scheme requires the SR image to improve its degradation estimate, the predictor model needs to be trained simultaneously with the SR model. We used the same CNN-based predictor network described in DANv1 [23] for our models and fixed the iteration count to four in all cases (matching the implementation as described in [23]). We coupled this predictor with our non-blind SR models using the framework described in Section 3.2. We trained all DAN models by optimising for the SR L1 loss (identical to the non-blind models) and an additional L1 loss component comparing the prediction and ground-truth vectors. Target vectors varied according to the pipeline, the details of each are provided in their respective results sections. For each specific model architecture, the hyperparameters and validation selection criteria were all set to be identical to that of the base, non-blind model. The batch size for all models was adjusted to 4 due to the increased GPU memory requirements needed for the iterative training scheme. Accordingly, whenever a warm restart scheduler was used, the restart point was adjusted to 250,000 iterations (to maintain the same total number of iterations as performed by the other models that utilised a batch size of 8 for 125,000 iterations).

Additionally, we also trained the original DANv1 model from scratch, using the same hyperparameters from [23] and the same validation scheme as the other DAN models. The batch size was also fixed to 4 in all cases.

- **Contrastive Learning**: We used the same encoder from [24] for most of our contrastive learning schemes. This encoder consists of a convolutional core connected to a set of three fully-connected layers. During training, we used the output of the fully-connected layers ($Q$) to calculate loss values (i.e., $f_q$ and $f_k$ in Section 5) and update the encoder weights, following [24]. Before coupling the encoder with an SR network, we first pre-trained the encoder directly. For this pre-training, the batch size was set to 32 and data was generated online, i.e., each LR image was synthesised on the fly at runtime. All encoders were trained with a constant learning rate of $1 \times 10^{-3}$, a patch size of $64 \times 64$ and the Adam optimiser. The encoders were trained until the loss started to plateau and t-Distributed Stochastic Neighbour Embedding (t-SNE) clustering of degradations generated on a validation set composed of 400 images from CelebA [89] and BSDS200 [81] was clearly visible (more details on this process are provided in Section 4.3.1). In all cases, the temperature hyperparameter, momentum value, queue length, and encoder output vector size were set to 0.07, 0.999, 8192 and 256, respectively, (matching the models from [24]).

After pre-training, each encoder was coupled to non-blind SR networks using the framework discussed in Section 3.2. For standard encoders, the encoding (i.e., the output from the convolutional core that bypasses the fully-connected layers) is typically fed into metadata insertion blocks directly, unless specified. For encoders with a regression component (see Figure 2B), the dropout output is fed to the metadata insertion block instead of the encoding. The combined encoder + SR network was then trained using the same dataset and hyperparameters as the non-blind case. The encoder weights were frozen and no gradients were generated for the encoding at runtime, unless specified.
In our analysis, we use the simple pipeline as our primary blind SR task and the complex pipeline as an extension scenario for the best performing methods. In Section 4.2 we discuss our metadata insertion block testing, while Sections 4.3 and 4.4 present our degradation prediction and SR analysis on the simple pipeline respectively. Sections 4.5 and 4.6 follow up with our analysis on the complex pipeline and Section 4.7 presents some of our blind SR results on real-world degraded images.

4.2. Metadata Insertion Block Testing

To test and compare the various metadata insertion blocks selected, we implemented each block into RCAN, and trained a separate model from scratch on our simple pipeline dataset. Each metadata insertion block was given either the real blur kernel width (normalised in the range \([0, 1]\)) or the PCA-reduced kernel representation, for each LR image. The PSNR test results for each model have been compiled in Table 1, and plotted in a comparative bar graph in Figure 5. The SSIM results are also available in the supplementary information (Table S1).

Table 1. PSNR (dB) SR results on simple pipeline comparing metadata insertion blocks. The labels 'low', 'med' and 'high' refer to \(\sigma\) values of 0.2, 1.6, and 3.0, respectively. Models in the 'Non-Blind' category are all RCAN models, upgraded with one instance of the indicated metadata insertion block, unless stated otherwise. 'MA (all)' and 'DA (all)' refer to RCAN with 200 individual MA or DA blocks inserted throughout the network. 'MA (PCA)' refers to RCAN with a single MA block provided with a 10-element PCA-reduced vector of the blur kernel applied. 'DGFMB (no FC)' refers to RCAN with a DGFMB layer where the metadata input is not passed through a fully-connected layer prior to concatenation (refer to Figure 2E). The best result for each set is shown in red, while the second-best result is shown in blue.

| Model | Set5 Low | Med | High | Set14 Low | Med | High | BSDS100 Low | Med | High | Manga109 Low | Med | High | Urban100 Low | Med | High |
|-------|---------|-----|------|----------|-----|------|-------------|-----|------|--------------|-----|------|--------------|-----|------|
| Baselines |         |     |      |          |     |      |              |     |      |               |     |      |              |     |      |
| Bicubic | 27.084  | 25.857 | 23.867 | 24.532  | 23.695 | 22.286 | 24.647       | 23.998 | 22.910 | 23.608        | 22.564 | 21.805 | 21.104        | 19.944 |
| Lanczos | 27.462  | 26.210 | 24.039 | 24.760  | 23.925 | 22.409 | 24.811       | 24.173 | 23.007 | 23.923        | 22.850 | 21.071 | 21.293        | 20.046 |
| RCAN    | 30.675  | 30.484 | 29.635 | 30.007  | 29.003 | 26.149 | 26.337       | 26.379 | 25.898 | 29.206        | 29.406 | 24.962 | 24.899        | 23.960 |
| Non-Blind |       |     |      |          |     |      |              |     |      |               |     |      |              |     |      |
| MA      | 30.956  | 30.973 | 29.880 | 27.091  | 27.094 | 26.383 | 26.389       | 26.440 | 26.002 | 29.529        | 29.834 | 28.799 | 25.059        | 25.088 | 24.218 |
| MA (all) | 30.926  | 30.911 | 29.911 | 27.057  | 27.086 | 26.373 | 26.386       | 26.441 | 26.005 | 29.542        | 29.874 | 28.865 | 25.043        | 24.995 | 24.224 |
| MA (PCA) | 30.947  | 30.942 | 29.873 | 27.052  | 27.038 | 26.382 | 26.388       | 26.442 | 26.003 | 29.489        | 29.817 | 28.740 | 25.077        | 25.014 | 24.227 |
| SRMD    | 30.946  | 30.960 | 29.906 | 27.086  | 27.072 | 26.358 | 26.393       | 26.439 | 25.998 | 29.515        | 29.806 | 28.750 | 25.072        | 25.012 | 24.204 |
| SFT     | 30.960  | 30.958 | 29.972 | 27.065  | 27.066 | 26.386 | 26.388       | 26.439 | 26.006 | 29.541        | 29.849 | 28.835 | 25.088        | 25.022 | 24.241 |
| DA      | 30.930  | 30.934 | 29.929 | 27.088  | 27.093 | 26.389 | 26.392       | 26.449 | 26.009 | 29.515        | 29.839 | 28.801 | 25.056        | 25.019 | 24.239 |
| DA (all) | 30.956  | 30.958 | 29.888 | 27.033  | 27.044 | 26.358 | 26.378       | 26.437 | 25.990 | 29.451        | 29.844 | 28.781 | 24.998        | 24.999 | 24.207 |
| DGFMB   | 30.969  | 30.955 | 29.891 | 27.068  | 27.060 | 26.347 | 26.386       | 26.440 | 26.001 | 29.508        | 29.811 | 28.802 | 25.067        | 25.007 | 24.228 |
| DGFMB (no FC) | 30.985 | 30.941 | 29.909 | 27.062  | 27.077 | 26.388 | 26.392       | 26.436 | 26.006 | 29.508        | 29.806 | 28.781 | 25.073        | 25.014 | 24.237 |

From the results, it is evident that metadata insertion provides a significant boost to performance across the board. Somewhat surprisingly, the results also show that no single metadata insertion block has a clear advantage over the rest. Every configuration tested, including those where multiple metadata insertion blocks are provided, produces roughly the same level of performance with only minor variations across dataset/degradation combinations. This outcome suggests that each metadata block is producing the same amount of useful information from the input kernel. Further complexity, such as the DA block’s kernel transformation or the SFT/DGFMB feature map concatenation, provides no further gain in performance. Even adding further detail to the metadata, such as by converting the full blur kernel into a PCA-reduced vector, provides no performance gains. This again seems to suggest that the network is capable of extrapolating the kernel width to
the full kernel description, without requiring any additional data engineering. Furthermore, adding just a single block at the beginning of the network appears to be enough to inform the whole network, with additional layers providing no improvement (while a decrease in performance is actually observed in the case of DA). We hypothesise that this might be due to the fact that degradations are mostly resolved in the earlier low-frequency stages of the network.

Given that all metadata insertion blocks provide almost identical performance, we selected a single MA block for our blind SR testing, given its low overhead and simplicity with respect to the other approaches. While it is clear the more complex metadata insertion blocks do not provide increased performance on this dataset, it is still possible that they might provide further benefit if other types of metadata are available.

4.3. Blur Kernel Degradation Prediction

To test our degradation prediction mechanisms, we evaluated the performance of these methods on a range of conditions and datasets.

4.3.1. Contrastive Learning

For contrastive learning methods, the prediction vectors generated are not directly interpretable. This makes it difficult to quantify the accuracy of the prediction without some form of clustering/regression analysis. However, through the use of dimensionality reduction techniques such as t-SNE [90], the vectors can be easily reduced to 2-D, which provides an opportunity for qualitative screening of each model in the form of a graph.

We trained separate encoders using our three contrastive algorithms on the simple degradation pipeline. The testing epoch and training details for each model are provided in Table 2. For SupMoCo schemes, the models were trained with triple precision labelling (low/medium/high labels) with respect to the blur $\sigma$ value. For WeakCon, the weighting
was found by calculating the Euclidean distance between the normalised kernel widths (\( \sigma \)) of the query and negative samples (as proposed in [25]). The number of training epochs for each encoder was selected qualitatively, based on when the contrastive loss started to plateau, and after degradation clustering could be qualitatively observed on the validation set. Training the encoders beyond this point seemed to provide little to no performance gain, as is shown in Table 3.

We used the trained encoders to generate prediction vectors for the entirety of the BSDS100, Manga109 and Urban100 testing datasets, with each image degraded with three different kernel widths (927 images in total), and then applied t-SNE reduction for each set of outputs. We also fed this test set to a DASR pretrained encoder (using the weights provided in [24]) for direct comparison with our own networks. The t-SNE results are presented in Figure 6.

It is immediately apparent that all of the models achieve some level of separation between the three different \( \sigma \) values. However, the semi-supervised methods produce very clear clustering (with just a few outliers) while the MoCo methods generate clusters with less well-defined edges. The influence of the labelling systems clearly produces a very large repulsion effect between the different \( \sigma \) widths, which the unsupervised MoCo system cannot match. Interestingly, there is no discernable distinction between the WeakCon and SupMoCo plots, despite their different modes of action. Additionally, minor modifications to the training process such as swapping the encoder for a larger model (e.g., ResNet) or continuing to train the predictor in tandem with an SR model (SR results in Section 4.4) appear to provide no benefit or even degrade the output clusters.

![Figure 6. t-SNE plots (perplexity value of 40) showing the separation power of the different contrastive learning algorithms considered. Each dimension was independently normalised in the range [0, 1] after computing the t-SNE results on the encoded vectors generated on our test set of 927 LR images.](image-url)
Table 2. Contrastive encoders considered in our ‘simple pipeline’ analysis. The ‘positive patches per query patch’ column refers to how many patches are produced per iteration to act as positive samples, as stipulated in Equations (6)–(8).

| Model                  | Positive Patches per Query Patch | Epoch Selected | Loss               |
|------------------------|----------------------------------|----------------|--------------------|
| MoCo                   | 1                                | 2104           | Equation (6)       |
| SupMoCo                | 3                                | 542            | Equation (7)       |
| WeakCon                | 1                                | 567            | Equation (8)       |
| SupMoCo (regression)   | N/A                              | 376            | L1 loss            |
| SupMoCo (contrastive + regression) | 3                  | 115            | Equation (7) & L1 loss |
| SupMoCo (ResNet)       | 3                                | 326            | Equation (7)       |

4.3.2. Regression Analysis

For our iterative and regression models, the output prediction is much simpler to interpret. Direct $\sigma$ and PCA kernel estimates can be immediately compared with the actual value. We trained a variety of iterative DAN models, using RCAN as our base SR model for consistency. Several separate RCAN-DAN models were implemented; one specifically predicting the $\sigma$ and others predicting a 10-element PCA representation of each kernel. We also trained two DANv1 models (predicting PCA kernels) from scratch for comparison: one using a fixed learning rate of $2 \times 10^{-4}$ (matching the original implementation in [23]) and one using our own cosine annealing scheduler with a restart value of 250,000 (matching our other models). We compared the prediction capabilities of our models, and a number of pretrained checkpoints from the literature, on our testing sets (blurred with multiple values of $\sigma$). The pretrained models for IKC, DANv1 and DANv2 were extracted from their respective official code repositories. These pretrained models were also trained on DIV2K/Flickr2K images, but training degradations were generated online (with $\sigma$ in the range $[0.2, 4]$), which should result in superior performance. Figure 7A shows the prediction error of the direct regression models that were trained (both contrastive and iterative models). The results clearly show that the DAN predictor is the strongest of those tested, with errors below 0.05 in some cases (representing an error of less than 2.5%). The contrastive/regression methods, while producing respectable results in select scenarios, seem to suffer across most of the distribution tested. In both types of models, the error seems to increase when the width is at its lower range. We hypothesise that, at this point, it is difficult to distinguish between $\sigma$ of 0.2–0.4, given that the corresponding kernels are quite small.

Figure 7B shows the results of the PCA prediction models. The plot shows that our RCAN-DAN models achieve very similar prediction performance to the pretrained DANs. What makes this result remarkable is the fact that our models were trained for much less time than the pretrained DAN models, both of which were trained for $\approx$7000 epochs. Training DANv1 from scratch for the same amount of time as our models (1000 epochs) shows that the prediction performance at this point is markedly worse. It is clear that the larger and more capable RCAN model is helping boost the $\sigma$ prediction performance significantly. On the other hand, the pretrained IKC model is significantly outclassed by all DAN models in almost all scenarios. It is also worth noting that the prediction of kernels at the lower end of the spectrum suffers from increased error, across the board.
4.4. Blind SR on Simple Pipeline

The real test for our combined SR and predictor models is the blind SR performance. Table 3 presents the blind SR PSNR results of all the models considered on the test sets under various levels of blur $\sigma$. SSIM results are also provided in the supplementary information (Table S2). Figures 8 and 9 further complement these results, with a bar chart comparison of key models and a closer look at the SR performance across various levels of $\sigma$, respectively.

With reference to the model categories highlighted in Table 3, we make the following observations:

- **Pretrained models**: We provide the results for the pretrained models evaluated in Figure 7B, along with the results for the pretrained DASR [24], a blind SR model with a MoCo-based contrastive encoder. The DAN models have the best results in most cases (with DANv2 having the best performance overall). For IKC, another iterative model, we present two sets of metrics: IKC (pretrained-best-iter) shows the results obtained when selecting the best image from all SR output iterations (7 in total), as is the implementation in the official IKC codebase. IKC (pretrained-last-iter) shows the results obtained when selecting the image from the last iteration (as is done for the DAN models). The former method produces the best results (even surpassing DAN in some cases), but cannot be applied in true blind scenarios where a reference HR image is not available.
• **Non-blind models**: The non-blind models fed with the true blur kernel width achieve the best performance of all the models studied. This is true both for RCAN and HAN, with HAN having a slight edge overall. The wide margin over all other blind models clearly shows that significantly improved performance is possible if the degradation prediction system can be improved. We also trained and tested a model (RCAN-MA (noisy sigma)) which was provided with the normalised $\sigma$ values corrupted by noise (mean 0, standard deviation 0.1). This error level is slightly higher than that of the DAN models tested (Figure 7A), allowing this model to act as a performance reference for our estimation methods.

• **RCAN-DAN models**: As observed in Figure 7, the DAN models that were trained from scratch are significantly worse than the RCAN models, including the fully non-blind RCAN, across all datasets and $\sigma$ values, as shown in Figures 8 and 9 respectively. The RCAN-DAN models show a consistent performance boost over RCAN across the board. As noted earlier in Section 4.2, predicting PCA-reduced kernels appears to provide no advantage over directly predicting the kernel width.

• **RCAN-Contrastive models**: For the contrastive models, the results are much less clear-cut. The different contrastive blind models exhibit superior performance to RCAN under most conditions (except for Urban100), but none of the algorithms tested (MoCo, SupMoCo, WeakCon and direct regression) seem to provide any particular advantage over each other. The encoder trained with combined regression and SupMoCo appears to provide a slight boost over the other techniques (Figure 8), but this is not consistent across the datasets and $\sigma$ values analysed. This is a surprising result, given that the clear clusters formed by SupMoCo and WeakCon (as shown in Figure 6) would have been expected to improve the encoders’ predictive power. We hypothesise that the encoded representation is difficult for even deep learning models to interpret, and a clear-cut route from the encoded vector to the actual blur $\sigma$ may be difficult to produce. We also observe that both the RCAN-DAN and RCAN-SupMoCo models clearly surpass the noisy sigma non-blind RCAN model on datasets with medium and high $\sigma$, while they perform slightly worse on datasets with low $\sigma$. This matches the results in Figure 7, where it is clear that the performance of all predictors suffer when $\sigma$ is low.

• **HAN models**: Upgraded HAN models appear to follow similar trends as RCAN models. The inclusion of DAN provides a clear boost in performance, but this time the inclusion of the SupMoCo-regression predictor seems to only boost performance when $\sigma$ is high.

• **Extensions**: We also trained a RCAN-DAN model where we pre-initialised the predictor with that from the pretrained DANv1 model. The minor improvements indicate that, for the most part, the predictor is achieving similar prediction accuracy to that of the pretrained models (as is also indicated in Figure 7). We also extended the training of the baseline RCAN and the RCAN-SupMoCo-regression model to 2200 epochs. The expanded training continues to improve performance and, perhaps crucially, the contrastive model continues to show a margin of improvement over the baseline RCAN model. In fact, this extended model starts to achieve similar or better performance than the pretrained DAN models (shown both in Table 3 and Figure 9). This is achieved with a significantly shorter training time (2200 vs. $\approx 7000$ epochs) and a fixed set of degradations, indicating that our models would surpass the performance of the pretrained DAN models if trained with the same conditions.
Table 3. PSNR (dB) SR results on the simple pipeline comparing blind SR methods. The labels ‘low’, ‘med’ and ‘high’ refer to $\sigma$ values of 0.2, 1.6, and 3.0, respectively. For RCAN/HAN models with metadata, this is inserted using one MA block at the front of the network. ‘Noisy sigma’ refers to the use of normalised $\sigma$ metadata that has been corrupted with Gaussian noise (mean 0, standard deviation of 0.1) for both training and testing. ‘Long term’ refers to models that have been screened after training for 2200 epochs, rather than the standard 1000. The best result for each set is shown in red, while the second-best result is shown in blue.

| Model Set | Low | Med | High |
|-----------|-----|-----|------|
| Classical |     |     |      |
| Bicubic   | 27.084 | 25.857 | 23.867 |
| Lanczos   | 27.462 | 26.210 | 24.039 |
| Pretrained |     |     |      |
| IKC (pretrained-best-iter) | 30.828 | 30.494 | 30.049 |
| IKC (pretrained-last-iter) | 30.662 | 30.057 | 29.812 |
| DASR (pretrained) | 30.545 | 30.463 | 29.701 |
| DAN1 (pretrained) | 30.807 | 30.739 | 30.049 |
| DAN2 (pretrained) | 30.850 | 30.881 | 30.042 |
| Non-Blind |     |     |      |
| RCAN-MA (true sigma) | 30.956 | 30.973 | 28.790 |
| HAN-MA (true sigma) | 30.940 | 30.905 | 29.834 |
| RCAN-MA (noisy sigma) | 30.838 | 30.700 | 29.780 |
| RCAN/DAN |     |     |      |
| DANV1 | 30.432 | 30.387 | 29.436 |
| DANV1 (cosine) | 30.627 | 30.466 | 29.537 |
| RCAN (batch size 4) | 30.686 | 30.538 | 29.646 |
| RCAN-DAN | 30.813 | 30.827 | 29.741 |
| RCAN-DAN (sigma) | 30.747 | 30.666 | 29.782 |
| HAN/DAN |     |     |      |
| HAN (batch size 4) | 30.779 | 30.620 | 29.572 |
| HAN-DAN | 30.817 | 30.567 | 29.707 |
| RCAN/Contrastive |     |     |      |
| RCAN (batch size 8) | 30.675 | 30.484 | 29.635 |
| RCAN-MoCo | 30.870 | 30.677 | 29.714 |
| RCAN-SupMoCo | 30.819 | 30.700 | 29.720 |
| RCAN-regression | 30.730 | 30.618 | 29.642 |
| RCAN-SupMoCo-regression | 30.861 | 30.785 | 29.676 |
| RCAN-WeakConv | 30.741 | 30.600 | 29.729 |
| RCAN-SupMoCo (online) | 30.777 | 30.695 | 29.771 |
| HAN/Contrastive |     |     |      |
| HAN (batch size 8) | 30.705 | 30.592 | 29.593 |
| HAN-SupMoCo-regression | 30.734 | 30.659 | 29.741 |
| Extensions |     |     |      |
| RCAN-DAN (pretrained estimator) | 30.763 | 30.612 | 29.711 |
| RCAN (batch size 8, long-term) | 30.736 | 30.699 | 29.723 |
| RCAN-SupMoCo-regression (long-term) | 30.832 | 30.640 | 29.690 |

Additionally, we implemented, trained, and tested the Real-ESRGAN and ELAN models with the addition of the MA metadata insertion block (with the same hyperparameters as presented in Section 4.1). The testing results are available in the supplementary information (Table S3 containing Real-ESRGAN LPIPS results, and Tables S4 and S5 containing the PSNR and SSIM results for ELAN, respectively). For Real-ESRGAN, the addition of the true metadata (non-blind) makes a clear improvement over the base model. We also observed a consistent improvement in performance across datasets and $\sigma$ values for the DAN upgraded model. However, attaching the best performing SupMoCo encoder provided no clear advantage. We hypothesise that the Real-ESRGAN model is more sensitive to the accuracy of the kernel prediction, and thus sees limited benefit from the less accurate contrastive encoder (as we have shown for the DAN vs. contrastive methods (Figure 7)).
Figure 8. Bar graph showing improvement in PSNR over the pretrained DASR, for key models from Table 3. PSNR improvements are stacked on each other for each specific blur kernel \( \sigma \) to show cumulative PSNR gain across datasets. Our hybrid models can match and even surpass the pretrained DAN models, despite the large disparity in training time.

Figure 9. Plot showing the relationship of SR performance (as measured using PSNR) with the level of blurring within an image. PSNR was measured by running each model on the BSDS100, Manga109 and Urban100 datasets (309 images) degraded with each specified \( \sigma \) value. ‘L1 loss’ refers to the regression component added to our contrastive encoders (Figure 3). All models appear to show degraded performance with higher levels of blurring, indicating that none of the models analysed are capable of fully removing blurring effects from the final image. The non-blind model outperforms all other models, indicating further improvements are possible with better degradation prediction.

For ELAN, the baseline model is very weak, and is actually surpassed by Lanczos upsampling in one case (both in terms of PSNR and SSIM). The addition of the true
metadata only appeared to help when MA was distributed through the whole network, upon which it increased the performance of the network massively (>3 dB in some cases). It is clear that ELAN does not perform well on these blurred datasets (ELAN was originally tested only on bicubically downsampled datasets). However, MA still appears to be able to significantly improve the model’s performance under the right conditions. Further investigation is required to first adapt ELAN for such degraded datasets before attempting to use this model as part of our blind framework.

4.5. Complex Degradation Prediction

For our extended analysis on more realistic degradations, we trained three contrastive encoders (MoCo, SupMoCo and WeakCon) and one RCAN-DAN model on the complex pipeline dataset (Section 4.1). Given the large quantity of degradations, we devised a number of testing scenarios, each applied on the combined images of BSDS100, Manga109 and Urban100 (309 images total). The scenarios we selected are detailed in Table 4. We will refer to these testing sets for the rest of this analysis. We evaluated the prediction capabilities of the contrastive and iterative models separately. We purposefully limited the testing blur kernel shapes to isotropic/anisotropic Gaussians to simplify analysis.

Table 4. The different testing scenarios considered for the complex analysis. Each scenario was applied on all images of the BSDS100, Manga109 and Urban100 datasets (309 images total). Cases which include noise have double the amount of images (618), since the pipeline was applied twice: once with colour noise, and once with grey noise. The last scenario consists of every possible combination of the degradations considered (16 total combinations with 4944 images in total, including colour and grey noise). For all cases, isotropic blurring was applied with a $\sigma$ of 2.0, anisotropic blurring was applied with a horizontal $\sigma$ of 2.0, a vertical $\sigma$ of 1.0 and a random rotation, Gaussian/Poisson noise were applied with a sigma/scale of 20.0/2.0, respectively, and JPEG/JM H.264 compression were applied with a quality factor/QPI of 60/30, respectively. All scenarios also included $\times 4$ bicubic downsampling inserted at the appropriate point (following the sequence in Equation (3)).

| Test Scenario | Blurring         | Noise            | Compression       | Total Images |
|---------------|------------------|------------------|-------------------|--------------|
| JPEG          | N/A              | N/A              | JPEG             | 309          |
| JM            | N/A              | N/A              | JM H.264         | 309          |
| Poisson       | N/A              | Poisson          | N/A              | 618          |
| Gaussian      | N/A              | Gaussian         | N/A              | 618          |
| Iso           | Isotropic        | N/A              | N/A              | 309          |
| Aniso         | Anisotropic      | N/A              | N/A              | 309          |
| Iso + Gaussian| Isotropic        | Gaussian         | N/A              | 618          |
| Gaussian + JPEG| Isotropic     | Gaussian         | JPEG             | 618          |
| Iso + Gaussian + JPEG | Isotropic | Gaussian         | JPEG             | 618          |
| Aniso + Poisson + JM | Anisotropic | Poisson          | JM H.264         | 618          |
| Iso/Aniso + Gaussian/Poisson + JPEG/JM | Isotropic & Gaussian & | Poisson          | JPEG & JM       | 4944         |

4.5.1. Contrastive Learning

For each of the contrastive algorithms, we trained an encoder (all with the same architecture as used for the simple pipeline) with the following protocol:

- We first pre-trained the encoder with an online pipeline of noise (same parameters as the full complex pipeline, but with an equal probability to select grey or colour noise) and bicubic downsampling. We found that this pre-training helps reduce loss stagnation for the SupMoCo encoder, so we applied this to all encoders. The SupMoCo encoder was trained with double precision at this stage. We used 3 positive patches for SupMoCo and 1 positive patch for both MoCo and WeakCon.
• After 1099 epochs, we started training the encoder on the full online complex pipeline (Section 4.1). The SupMoCo encoder was switched to triple precision from this point onwards.

• We stopped all encoders after 2001 total epochs, and evaluated them at this checkpoint.

• For SupMoCo, the decision tree in Section 3.5.2 was used to assign class labels. For WeakCon, $w_{ij}$ was computed as the Euclidean distance between query/negative sample vectors containing: the vertical and horizontal blur $\sigma$, the Gaussian/Poisson sigma/scale, respectively, and the JPEG/JM H.264 quality factor/QPI, respectively, (6 elements total). All values were normalised to $[0, 1]$ prior to computation.

As with the simple pipeline, contrastive encodings are not directly interpretable and so we analysed the clustering capabilities of each encoder through t-SNE visualizations. We evaluated each encoder on the full testing scenario (Iso/Aniso + Gaussian/Poisson + JPEG/JM in Table 4), and applied t-SNE independently for each model. The results are shown in Figure 10.

![Figure 10](image_url)

Figure 10. t-SNE plots (perplexity value of 40) showing the separation power of the different contrastive learning algorithms considered on the complex pipeline. All models were evaluated on the Iso/Aniso + Gaussian/Poisson + JPEG/JM testing scenario having 4944 images (ref. Table 4). Each dimension was independently normalised in the range $[0, 1]$ after computing the t-SNE results. The data used by each method is identical across rows, with different colours pertaining to the degradation considered: (A) t-SNE plots with each point labelled according to the blur kernel applied. Only 560 images (randomly selected) are shown for each plot, to reduce cluttering. Arrows indicate the rotation of the anistropic kernels. (B) t-SNE plots with each point coloured according to the noise injected. (C) t-SNE plots with each point coloured according to the compression applied.

It is evident from the t-SNE plots that the clustering of the dataset is now significantly more complex than that observed in Figure 6. However, all three encoders appear to have successfully learnt how to distinguish between the two compression types and are also
mostly successful when clustering the four types of noise (MoCo is slightly weaker for grey noise). In the supplementary information (Figure S1), we also show that the encoders are capable of separating different intensities of both compression and noise, albeit with less separation of the two noise types.

For blurring, the separation between isotropic and anisotropic kernels is much less logical. It appears that each encoder was attempting to form sub-clusters for each type of kernel in some cases (in particular for SupMoCo) but the separation is significantly less clear cut than that obtained in Figure 6. Further analysis would be required to decipher whether clustering is weak simply due to the difficulty of the exercise, or whether clustering is being mostly influenced by the other degradations considered in the pipeline.

As observed with the simple pipeline, it is again apparent that the different methods of semi-supervision seem to be converging to similar results. This is also in spite of the fact that WeakCon was supplied with only 6 degradation elements while SupMoCo was supplied with the full degradation metadata through its class system. Further investigation into their learning process could reveal further insight into the effects of each algorithm.

4.5.2. Iterative Parameter Regression

The RCAN-DAN model was trained on the complex pipeline dataset with identical hyperparameters to that of the simple pipeline. For degradation prediction, we set the DAN model to predict a vector with the following elements (15 total):

- Individual elements for the following blur parameters: vertical and horizontal $\sigma$, rotation, individual $\beta$ for generalised Gaussian and plateau kernels and the sinc cutoff frequency. Whenever one of these elements was unused (e.g., cutoff frequency for Gaussian kernels), this was set to 0. All elements were normalised to $[0, 1]$ according to their respective ranges (Section 4.1).
- Four boolean (0 or 1) elements categorising whether the kernel shape was:
  - Isotropic or anisotropic
  - Generalised
  - Plateau-type
  - Sinc
- Individual elements for the Gaussian sigma and Poisson scale (both normalised to $[0, 1]$).
- A boolean indicating whether the noise was colour or grey type.
- Individual elements for the JM H.264 QPI and JPEG quality factor (both normalised to $[0, 1]$).

We tested the prediction accuracy by evaluating the model on a number of our testing scenarios, and then quantifying the degradation prediction error. The results are shown in Table 5. As observed with the contrastive models, blur kernel parameter prediction accuracy is extremely low, even when no other degradations are present. On the other hand, both noise and compression prediction are significantly better, with sub 0.1 error in all cases, even when all degradations are present. We hypothesise that since blur kernels are introduced as the first degradation in the pipeline, most of the blurring information could be masked when noise addition and compression have been applied.

To the best of our knowledge, we are the first to present fully explicit blind degradation prediction on this complex pipeline. We hope that the prediction results achieved in this analysis can act as a baseline from which further advances and improvements can be made.
Table 5. Mean L1 error for the DAN predictor within the RCAN-DAN model trained on the complex pipeline. Prediction accuracy is high for the compression and noise degradations, but weak for blur. Blur parameters incorporate all the blur-specific numerical parameters and the four booleans. Noise parameters incorporate the Gaussian sigma, Poisson scale and colour/grey boolean. Compression parameters incorporate the JPEG quality factor and JM H.264 QPI. The degradation scenarios tested here are described in Table 4.

| Test Scenario                  | Blurring (σ, kernel type) | Noise (scale, gray/colour type) | Compression (QPI/quality) |
|-------------------------------|---------------------------|---------------------------------|---------------------------|
| Iso/Aniso                     | 0.318                     | N/A                             | N/A                       |
| Gaussian/Poisson              | N/A                       | 0.042                           | N/A                       |
| JPEG/JM                       | N/A                       | N/A                             | 0.078                     |
| Iso/Aniso + Gaussian/Poisson  | 0.317                     | 0.036                           | 0.070                     |
| + JPEG/JM                     |                           |                                 |                           |

4.6. Blind SR on Complex Pipeline

For blind SR on the complex pipeline, we focus on just the RCAN and RCAN upgraded models to simplify analysis. We use a single MA block to insert metadata into the SR core in all cases apart from one, where we distribute MA throughout RCAN. We also trained a number of non-blind models (fed with different quantities of the correct metadata) as comparison points. PSNR SR results comparing the baseline RCAN to the blind models are provided in Table 6 (Table S6 in the Supplementary Information provides the SSIM results).

We make the following observations on these results:

- **Compression- and noise-only scenarios:** In these scenarios, the RCAN-DAN model shows clear improvement over all other baseline and contrastive encoders (apart from some cases on Manga109). Improvement is most significant in the compression scenarios.

- **Blur-only scenarios:** Since the blurring scenarios are very similar or identical to the simple pipeline, the models from Table 3 (RCAN is also shown in Table 6) are significantly stronger. The DAN model overtakes the baseline in some cases, but is very inconsistent.

- **Multiple combinations:** In the multiple degradation scenarios, the DAN model consistently overtakes the baselines, but PSNR/SSIM increases are minimal.

For all scenarios, there are a number of other surprising results. The contrastive methods appear to be providing no benefit to SR performance, in almost every case. Furthermore, the non-blind models are often overtaken by the DAN model in certain scenarios, and the amount of metadata available to the non-blind models does not appear to correlate with the final SR performance. It is clear that the metadata we have available for these degradations are having a much lesser impact on SR performance than on the simple pipeline. Since the contrastive encoders have shown to be slightly weaker than DAN in the simple pipeline case (Figure 7), it is clear that their limited prediction accuracy is also limiting potential gains in SR performance on this pipeline. This dataset is significantly more difficult than the simple case, not just due to the increased amount of degradations, but also as the models appear less receptive to the insertion of metadata. We again hope that these results can act as a baseline for further exploration into complex blind SR.
Table 6. PSNR (dB) SR results on the complex pipeline comparing blind SR methods. Metadata is inserted into RCAN models using one MA block at the front of the network, apart from models with the ‘all’ suffix where 200 independent MA blocks are inserted throughout the network. The degradation scenarios tested here are described in Table 4. The non-blind models were fed with a vector containing the vertical/horizontal blur ρ, the kernel type (7 possible shapes), the Gaussian sigma/Poisson scale, a boolean indicating the addition of grey or colour noise, and the JPEG quality factor/JM H.264 QPI. Non-blind models marked with a degradation (e.g., no blur) have the marked degradation parameters removed from their input vector. All values were normalised to [0, 1] apart from the kernel type. The best result for each set is shown in red, while the second-best result is shown in blue.

| Dataset | Model                         | JPEG  | JM    | Poisson | Gaussian | Iso | Aniso | Iso + Gaussian | Gaussian + JPEG | Iso + Gaussian + JPEG | Aniso + Poisson + JM | Iso/Aniso + Gaussian/Poisson + JPEG/JM |
|---------|-------------------------------|-------|-------|---------|----------|-----|-------|----------------|-----------------|---------------------|---------------------|--------------------------------------|
| BSDS100| RCAN (batch size 4)           | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (batch size 8)           | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind)              | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no blur)     | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no noise)    | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no compression) | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (MoCo)                   | 21.25 | 25.34 | 21.25   | 21.25    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (SupMoCo, all)           | 21.25 | 25.34 | 21.25   | 21.25    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN-DAN                      | 21.25 | 25.34 | 21.25   | 21.25    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
| Manga109| RCAN (batch size 4)           | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (batch size 8)           | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind)              | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no blur)     | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no noise)    | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no compression) | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (MoCo)                   | 21.61 | 26.24 | 21.61   | 21.61    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (SupMoCo, all)           | 21.61 | 26.24 | 21.61   | 21.61    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN-DAN                      | 21.61 | 26.24 | 21.61   | 21.61    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
| Urban100| RCAN (batch size 4)           | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (batch size 8)           | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind)              | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no blur)     | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no noise)    | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (non-blind, no compression) | 21.36 | 24.38 | 21.36   | 21.36    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (MoCo)                   | 21.61 | 26.24 | 21.61   | 21.61    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (SupMoCo, all)           | 21.61 | 26.24 | 21.61   | 21.61    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN-DAN                      | 21.61 | 26.24 | 21.61   | 21.61    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |
|         | RCAN (trained on simple pipeline) | 21.05 | 26.15 | 21.05   | 21.05    | N/A | N/A   | N/A            | N/A             | N/A                 | N/A                 | N/A/Aniso + Gaussian/Poisson + JPEG/JM |

Sensors 2023, 23, 419
31 of 39
4.7. Blind SR on Real LR Images

As a final test to compare models from both pipelines, we ran a select number of models on real-world images from RealSRSet [19]. These results are shown in Figure 11, with an additional image provided in the supplementary information (Figure S2). This qualitative inspection clearly show that models trained on the complex pipeline are significantly better at dealing with real-world degradations than simple pipeline models. Figure 11 shows that the complex pipeline models can remove compression artefacts, sharpen images and smoothen noise. In particular, the dog image shows that RCAN-DAN can deal with noise more effectively than the baseline RCAN. The simple pipeline model results are all very similar to each other, as none of them are capable of dealing with degradations other than isotropic blurring.

**Figure 11.** Comparison of the SR results for various models, on images from RealSRSet [19]. All simple pipeline models (marked as ‘iso’) and the pretrained DAN model are incapable of dealing with degradations such as noise or compression. The complex pipeline models (marked as ‘full-stack’) produce significantly improved results. These models can sharpen details (first image), remove compression artefacts (second image) and smoothen noise (third image).
4.8. Results Summary

Given the large quantity of analyses conducted, we provide a brief summary of the most significant results obtained in each section here:

- In Section 4.2, we show that all of the metadata insertion mechanisms tested provide roughly the same SR performance boost when feeding a large network such as RCAN with non-blind blurring metadata. Furthermore, adding repeated blocks through the network provides little to no benefit. Given this result, we propose MA as our metadata insertion block of choice, as it provides identical SR performance as the other options considered, with very low complexity. Other metadata blocks could prove optimal in other scenarios (such as other degradations or with other networks), which would require further systematic investigation to determine.

- Section 4.3 provides a comparison of the prediction performance of the different algorithms considered on the simple blur pipeline. The contrastive algorithms clearly cluster images by the applied blur kernel width, with the semi-supervised algorithms providing the most well-defined separation between different values. The regression and iterative mechanisms are capable of explicitly predicting the blur kernel width with high accuracy, except at the lower extreme. Our prediction mechanisms combined with RCAN match the performance of the pretrained DAN models with significantly less training time.

- Section 4.4 compares the testing results of blind models built with our framework with baseline models from the literature. Each prediction mechanism considered elevates RCAN’s SR performance above its baseline value, for both PSNR and SSIM. In particular, the iterative mechanism provided the largest performance boost. For more complex models such as HAN and Real-ESRGAN, contrastive methods provide less benefit, but the iterative mechanism still shows clear improvements. Our models significantly overtake the SOTA blind DAN network when trained for the same length of time. In addition, our models approach or surpass the performance of the pretrained DANv1 and DANv2 checkpoints provided by their authors, which were trained for a significantly longer period of time.

- In Sections 4.5 and 4.6, we modify our prediction mechanisms to deal with a more complex pipeline of blurring, noise and compression, and attach these to the RCAN network. We show that the contrastive predictors can reliably cluster compression and noise, but blur kernel clustering is significantly weaker. Similarly, the iterative predictors are highly accurate when predicting compression/noise parameters, but are much less reliable for blur parameters. When testing their SR performance, the contrastive encoders seem to provide little to no benefit to RCAN’s performance. On the other hand, the DAN models reliably improve the baseline performance across various scenarios, albeit with limited improvements when all degradations are present at once. We anticipate that performance can be significantly improved with further advances to the prediction mechanisms and consider our results as a baseline for further exploration.

- Section 4.7 showcases the results of our models when applied to real-world LR images. Our complex pipeline models produce significantly better results than the pretrained DAN models and are capable of reversing noise, compression, and blurring in various scenarios.

5. Conclusions

In this work, a framework for combining degradation prediction systems with any SR network was proposed. By using a single metadata insertion block to influence the feature maps of a convolutional layer, a degradation vector from a prediction model can, in many cases, be used to improve the performance of the SR network. This premise was tested by implementing several contrastive and iterative degradation prediction mechanisms and coupling them with high-performing SR architectures. When tested on a dataset of images that were degraded by Gaussian blurring and downsampling, we show that our blind
mechanisms achieve at least the same (or better) blur σ prediction accuracy as the original methods, but with significantly less training time. Moreover, both blind degradation performance (in combined training cases, such as with DAN) and SR performance are substantially improved through the use of larger and stronger networks such as RCAN [8] or HAN [10]. Our results show that our hybrid models surpass the performance of the baseline non-blind and blind models under the same conditions. Other SR architecture categories such as the SOTA perceptual-loss based Real-ESRGAN [16] and the transformer-based ELAN architecture [13] also work within our framework, but the performance of these methods is more sensitive to the accuracy of the degradation prediction and the dataset used for training. We show that this premise also holds true for blind SR of a more complex pipeline involving various blurring, noise injection, and compression operations.

Our framework should enable blind SR research to be significantly expedited, since researchers can now focus their efforts on their degradation prediction mechanisms, rather than on deriving a custom SR architecture for each new method. There are various future avenues that could be explored to further assess the applications of our framework. Apart from investigating new combinations of blind prediction, metadata insertion and SR architectures, our framework could also be applied to new types of metadata. For example, blind prediction systems could be replaced with image classification systems. This will provide SR architecture with details on the image content (e.g., facial features for face SR [36]). Furthermore, the framework can be extended to video SR [91] where additional sources of metadata are available, such as the number of frames to be used in the super-resolution of a given frame as well as other details on the compression scheme, such as P- and B-frames (in addition to I-frames as considered in this work).

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/s23010419/s1, Table S1: SSIM SR results for metadata insertion block comparison; Table S2: SSIM SR results for simple pipeline comparison; Table S3: LPIPS SR results for Real-ESRGAN models on the simple pipeline; Table S4: PSNR results for ELAN models on the simple pipeline; Table S5: SSIM results for ELAN models on the simple pipeline; Table S6: SSIM SR results for complex pipeline comparison; Figure S1: additional t-SNE plots for complex pipeline contrastive encoders; Figure S2: Additional blind SR results on RealSRSet.

Author Contributions: Conceptualization, all authors; methodology, M.A. and K.G.C.; software, M.A., K.G.C. and C.G.; validation, M.A. and K.G.C.; formal analysis, M.A. and K.G.C.; investigation, M.A. and K.G.C.; resources, K.P.C., R.A.F. and J.A.; data curation, M.A. and K.G.C.; writing—original draft preparation, M.A., K.G.C. and C.G.; writing—review and editing, all authors; visualization, M.A. and K.G.C.; supervision, K.P.C., R.A.F. and J.A.; project administration, M.A., K.P.C. and J.A.; funding acquisition, R.A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research work forms part of the Deep-FIR project, which is financed by the Malta Council for Science & Technology (MCST), for and on behalf of the Foundation for Science & Technology, through the FUSION: R&I Technology Development Programme, grant number R&I-2017-002-T.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All code, data and model weights for the analysis presented in this paper are available here: https://github.com/um-dsrg/RUMpy, accessed on 27 December 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Gupta, R.; Sharma, A.; Kumar, A. Super-Resolution using GANs for Medical Imaging. *Procedia Comput. Sci.* 2020, 173, 28–35. [CrossRef]
2. Ahmad, W.; Ali, H.; Shah, Z.; Azmat, S. A new generative adversarial network for medical images super resolution. *Sci. Rep.* 2022, 12, 9533. [CrossRef] [PubMed]
3. Haut, J.M.; Fernandez-Beltran, R.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Pla, F. A New Deep Generative Network for Unsupervised Remote Sensing Single-Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* 2018, 56, 6792–6810. [CrossRef]
4. Wang, P.; Bayram, B.; Sertel, E. A comprehensive review on deep learning based remote sensing image super-resolution methods. Earth-Sci. Rev. 2022, 232, 104110. [CrossRef]
5. Zhang, J.; Xu, T.; Li, J.; Jiang, S.; Zhang, Y. Single-Image Super Resolution of Remote Sensing Images with Real-World Degradation Modeling. Remote Sens. 2022, 14, 2895. [CrossRef]
6. Chen, H.; He, X.; Qing, L.; Wu, Y.; Ren, C.; Sheriff, R.E.; Zhu, C. Real-world single image super-resolution: A brief review. Inf. Fusion 2022, 79, 124–145. [CrossRef]
7. Rasti, P.; Uiboupin, T.; Escalera, S.; Anbarjafari, G. Convolutional Neural Network Super Resolution for Face Recognition in Surveillance Monitoring. In Proceedings of the Articulated Motion and Deformable Objects, Palma de Mallorca, Spain, 13–15 July 2016; Perales, F.J., Kittler, J., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 175–184. [CrossRef]
8. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 294–310. [CrossRef]
9. Dai, T.; Cai, J.; Zhang, Y.; Xia, S.T.; Zhang, L. Second-Order Attention Network for Single Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 11057–11066. [CrossRef]
10. Niu, B.; Wen, W.; Ren, W.; Zhang, X.; Yang, L.; Wang, S.; Zhang, K.; Cao, X.; Shen, H. Single Image Super-Resolution via a Holistic Attention Network. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 191–207. [CrossRef]
11. Vella, M.; Mota, J.F.C. Robust Single-Image Super-Resolution via CNNs and TV-TV Minimization. IEEE Trans. Image Process. 2021, 30, 7830–7841. [CrossRef]
12. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 1833–1844. [CrossRef]
13. Zhang, X.; Zeng, H.; Guo, S.; Zhang, L. Efficient Long-Range Attention Network for Image Super-Resolution. In Proceedings of the Computer Vision—ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 649–667. [CrossRef]
14. Ledig, C.; Theis, I.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 105–114. [CrossRef]
15. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In Proceedings of the Computer Vision—ECCV 2018 Workshops, Munich, Germany, 8–14 September 2018; Leal-Taixé, L., Roth, S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 63–79. [CrossRef]
16. Wang, X.; Xie, L.; Dong, C.; Shan, Y. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 1905–1914. [CrossRef]
17. Zhang, W.; Shi, G.; Liu, Y.; Dong, C.; Wu, X.M. A Closer Look at Blind Super-Resolution: Degradation Models, Baselines, and Performance Upper Bounds. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022; pp. 527–536. [CrossRef]
18. Liu, A.; Liu, Y.; Gu, J.; Qiao, Y.; Dong, C. Blind Image Super-Resolution: A Survey and Beyond. IEEE Trans. Pattern Anal. Mach. Intell. 2022, 1–19. [CrossRef]
19. Zhang, K.; Liang, J.; Van Gool, L.; Timofte, R. Designing a Practical Degradation Model for Deep Blind Image Super-Resolution. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4791–4800. [CrossRef]
20. Jiang, J.; Wang, C.; Liu, X.; Ma, J. Deep Learning-Based Face Super-Resolution: A Survey. ACM Comput. Surv. 2021, 55, 1–36. [CrossRef]
21. Köhler, T.; Bätz, M.; Naderi, F.; Kaup, A.; Maier, A.; Riess, C. Toward Bridging the Simulated-to-Real Gap: Benchmarking Super-Resolution on Real Data. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 42, 2944–2959. [CrossRef]
22. Gu, J.; Lu, H.; Zuo, W.; Dong, C. Blind Super-Resolution With Iterative Kernel Correction. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1604–1613. [CrossRef]
23. Luo, Z.; Huang, Y.; Li, S.; Wang, L.; Tan, T. Unfolding the Alternating Optimization for Blind Super Resolution. In Proceedings of the Advances in Neural Information Processing Systems 33, Virtual, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 5632–5643. [CrossRef]
24. Wang, L.; Wang, Y.; Dong, C.; Xu, Q.; Yang, J.; An, W.; Guo, Y. Unsupervised Degradation Representation Learning for Blind Super-Resolution. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10576–10585. [CrossRef]
25. Zhang, Y.; Dong, L.; Yang, H.; Qings, L.; He, X.; Chen, H. Weakly-supervised contrastive learning-based implicit degradation modeling for blind image super-resolution. Knowl.-Based Syst. 2022, 249, 108984. [CrossRef]
26. Aquilina, M.; Galea, C.; Abela, J.; Camilleri, K.P.; Farrugia, R.A. Improving Super-Resolution Performance Using Meta-Attention Layers. *IEEE Signal Process. Lett.* 2021, 28, 2082–2086. [CrossRef]

27. Luo, Z.; Huang, H.; Yu, L.; Li, Y.; Fan, H.; Liu, S. Deep Constrained Least Squares for Blind Image Super-Resolution. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 17642–17652. [CrossRef]

28. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a Deep Convolutional Network for Image Super-Resolution. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 184–199. [CrossRef]

29. Bulat, A.; Tzimiropoulos, G. Super-FAN: Integrated Facial Landmark Localization and Super-Resolution of Real-World Low Resolution Faces in Arbitrary Poses with GANs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 109–117. [CrossRef]

30. Huang, H.; He, R.; Sun, Z.; Tan, T. Wavelet Domain Generative Adversarial Network for Multi-scale Face Hallucination. *Int. J. Comput. Vis.* 2019, 127, 763–784. [CrossRef]

31. Yu, X.; Fernando, B.; Ghanem, B.; Porikli, F.; Hartley, R. Face Super-Resolution Guided by Facial Component Heatmaps. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 219–235. [CrossRef]

32. Chen, Y.; Tai, Y.; Liu, X.; Shen, C.; Yang, J. FSRNet: End-to-End Learning Face Super-Resolution with Facial Priors. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2492–2501. [CrossRef]

33. Huang, H.; He, R.; Sun, Z.; Tan, T. Wavelet-SRNet: A Wavelet-Based CNN for Multi-scale Face Super Resolution. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1698–1706. [CrossRef]

34. Luo, Z.; Jiang, X.; Kot, A. Deep Coupled ResNet for Low-Resolution Face Recognition. *IEEE Signal Process. Lett.* 2018, 25, 526–530. [CrossRef]

35. Cao, Q.; Lin, L.; Shi, Y.; Liang, X.; Li, G. Attention-Aware Face Hallucination via Deep Reinforcement Learning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1656–1664. [CrossRef]

36. Yu, X.; Fernando, B.; Hartley, R.; Porikli, F. Super-Resolving Very Low-Resolution Face Images with Supplementary Attributes. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 908–917. [CrossRef]

37. Nguyen, N.L.; Anger, J.; Davy, A.; Arias, P.; Facciolo, G. Self-Supervised Super-Resolution for Multi-Exposure Push-Frame Satellites. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 1848–1858. [CrossRef]

38. Luo, Z.; Huang, Y.; Li, S.; Wang, L.; Tan, T. End-to-end Alternating Optimization for Blind Super Resolution. *arXiv* 2021. [CrossRef]

39. Zhang, K.; Zuo, W.; Zhang, L. Learning a Single Convolutional Super-Resolution Network for Multiple Degradations. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3262–3271. [CrossRef]

40. Xiao, J.; Yong, H.; Zhang, L. Degradation Model Learning for Real-World Single Image Super-Resolution. In Proceedings of the Computer Vision—ACCV 2020, Kyoto, Japan, 30 November–4 December 2020; Ishikawa, H., Liu, C.L., Pajdla, T., Shi, J., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 84–101. [CrossRef]

41. Yue, Z.; Zhao, Q.; Xie, J.; Zhang, L.; Meng, D.; Wong, K.Y.K. Blind Image Super-Resolution With Elaborate Degradation Modeling on Noise and Kernel. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 2128–2138. [CrossRef]

42. Emad, M.; Peemen, M.; Corporal, H. MoESR: Blind Super-Resolution using Kernel-Aware Mixture of Experts. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 4009–4018. [CrossRef]

43. Kang, X.; Li, J.; Duan, P.; Ma, F.; Li, S. Multilayer Degradation Representation-Guided Blind Super-Resolution for Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* 2022, 60, 1–12. [CrossRef]

44. Liu, P.; Zhang, H.; Cao, Y.; Liu, S.; Ren, D.; Zuo, W. Learning cascaded convolutional networks for blind single image super-resolution. *Neurocomputing* 2020, 417, 371–383. [CrossRef]

45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]

46. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced Deep Residual Networks for Single Image Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1132–1140. [CrossRef]

47. Li, W.; Lu, X.; Qian, S.; Lu, J.; Zhang, X.; Jia, J. On Efficient Transformer and Image Pre-training for Low-level Vision. *arXiv* 2021. [CrossRef]
48. Lu, Z.; Li, J.; Liu, H.; Huang, C.; Zhang, L.; Zeng, T. Transformer for Single Image Super-Resolution. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022; pp. 456–465. [CrossRef]
49. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 9992–10002. [CrossRef]
50. Chen, X.; Wang, X.; Zhou, J.; Dong, C. Activating More Pixels in Image Super-Resolution Transformer. arXiv 2022. [CrossRef]
51. Ha, V.K.; Ren, J.C.; Xu, X.Y.; Zhao, S.; Xie, G.; Masero, V.; Hussain, A. Deep Learning Based Single Image Super-Resolution: A Survey. Int. J. Autom. Comput. 2019, 16, 413–426. [CrossRef]
52. Wang, Z.; Chen, J.; Hoi, S.C.H. Deep Learning for Image Super-Resolution: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 2021, 43, 3365–3387. [CrossRef] [PubMed]
53. Xu, Y.S.; Tseng, S.Y.R.; Tseng, Y.; Kuo, H.K.; Tsai, Y.M. Unified Dynamic Convolutional Network for Super-Resolution With Variational Degradations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12493–12502. [CrossRef]
54. Cornillère, V.; Djelouah, A.; Yifan, W.; Sorkine-Hornung, O.; Schroers, C. Blind Image Super-Resolution with Spatially Variant Degradations. ACM Trans. Graph. 2019, 38. [CrossRef]
55. Yin, G.; Wang, W.; Yuan, Z.; Ji, W.; Yu, D.; Sun, S.; Chua, T.S.; Wang, C. Conditional Hyper-Network for Blind Super-Resolution With Multiple Degradations. IEEE Trans. Image Process. 2022, 31, 3949–3960. [CrossRef]
56. Kim, S.Y.; Sim, H.; Kim, M. KOALAnet: Blind Super-Resolution using Kernel-Oriented Adaptive Local Adjustment. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10606–10615. [CrossRef]
57. Bell-Kligler, S.; Shoher, A.; Irani, M. Blind Super-Resolution Kernel Estimation Using an Internal-GAN. In Proceedings of the Advances in Neural Information Processing Systems 32, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 284–293.
58. Shoher, A.; Cohen, N.; Irani, M. Zero-Shot Super-Resolution Using Deep Internal Learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3118–3126. [CrossRef]
59. Yuan, Y.; Liu, S.; Zhang, J.; Zhang, Y.; Dong, C.; Lin, L. Unsupervised Image Super-Resolution Using Cycle-in-Cycle Generative Adversarial Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 814–823. [CrossRef]
60. Zhou, Y.; Deng, W.; Tong, T.; Gao, Q. Guided Frequency Separation Network for Real-World Super-Resolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1722–1731. [CrossRef]
61. Maeda, S. Unpaired Image Super-Resolution Using Pseudo-Supervision. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 288–297. [CrossRef]
62. Bulat, A.; Yang, J.; Tzimiropoulos, G. To Learn Image Super-Resolution, Use a GAN to Learn How to Do Image Degradation First. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 187–202. [CrossRef]
63. Fritsche, M.; Gu, S.; Timofte, R. Frequency Separation for Real-World Super-Resolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, South Korea, 27–28 October 2019; pp. 3599–3608. [CrossRef]
64. Majumder, O.; Ravichandran, A.; Maji, S.; Achille, A.; Polito, M.; Soatto, S. Supervised Momentum Contrastive Learning for Few-Shot Classification. arXiv 2021.
65. Doersch, C.; Gupta, A.; Zisserman, A. CrossTransformers: Spatially-aware few-shot transfer. In Proceedings of the Advances in Neural Information Processing Systems 33, Virtual, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 21981–21993.
66. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised Contrastive Learning. In Proceedings of the Advances in Neural Information Processing Systems 33, Virtual, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 18661–18673.
67. Zhang, Z.; Sabuncu, M.R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In Proceedings of the Advances in Neural Information Processing Systems 31, Montreal, QC, Canada, 2–8 December 2018; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31, pp. 8792–8802.
68. Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; Fergus, R. Training Convolutional Networks with Noisy Labels. arXiv 2014.
69. Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; Bengio, S. Large Margin Deep Networks for Classification. In Proceedings of the Advances in Neural Information Processing Systems 31, Montreal, QC, Canada, 2–8 December 2018; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31, pp. 850–860.

70. Cao, K.; Wei, C.; Gaidon, A.; Archeda, N.; Ma, T. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In Proceedings of the Advances in Neural Information Processing Systems 32, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 1567–1578.

71. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-Margin Softmax Loss for Convolutional Neural Networks. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 20–22 June 2019; Balcan, M.F., Weinberger, Eds.; PMLR: Cambridge, MA, USA, 2016; Volume 48, pp. 507–516.

72. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the 37th International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020; PMLR: Cambridge, MA, USA, 2020; Volume 119, pp. 1597–1607.

73. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9726–9735. [CrossRef]

74. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved Baselines with Momentum Contrastive Learning. arXiv 2020. [CrossRef]

75. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. arXiv 2018. [CrossRef]

76. Sühling, K.; Tourapis, A.M.; Leontaris, A.; Sullivan, G. H.264/14496-10 AVC Reference Software Manual (revised for JM 19.0). 2015. Available online: http://iphome.hhi.de/suehring/tml/ (accessed on 2 July 2021).

77. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1122–1131. [CrossRef]

78. Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.H.; Zhang, L. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1100–1121. [CrossRef]

79. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Morel, M.A. Low-Complexity Single-Image Super-Resolution Based on Nonnegative Neighbor Embedding. In Proceedings of the British Machine Vision Conference, Surrey, UK, 3–7 September 2012; Bowden, R., Collomosse, J., Mikolajczyk, K., Eds.; BMVA Press: Durham, UK, 2012; pp. 135.1–135.10. [CrossRef]

80. Zeyde, R.; Elad, M.; Protter, M. On Single Image Scale-Up Using Sparse-Representations. In Proceedings of the Curves and Surfaces, Avignon, France, 24–30 June 2010; Boissonnat, J.D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Masure, M.L., Schumaker, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 711–730. [CrossRef]

81. Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2011, 33, 898–916. [CrossRef]

82. Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Sketch-Based Manga Retrieval Using Manga109 Dataset. Multimed. Tools Appl. 2017, 76, 21811–21838. [CrossRef]

83. Huang, J.B.; Singh, A.; Ahuja, N. Single Image Super-Resolution from Transformed Self-Exemplars. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5206. [CrossRef]

84. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans. Image Process. 2004, 13, 600–612. [CrossRef]

85. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595. [CrossRef]

86. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems 32, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.

87. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.

88. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

89. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3730–3738. [CrossRef]
90. van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

91. Liu, H.; Ruan, Z.; Zhao, P.; Dong, C.; Shang, F.; Liu, Y.; Yang, L.; Timofte, R. Video super-resolution based on deep learning: A comprehensive survey. *Artif. Intell. Rev.* **2022**, *55*, 5981–6035. [CrossRef]

**Disclaimer/Publisher’s Note**: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.