CAMERA SETTINGS PREDICTION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Edward Laurence[1] *& Charles Murphy*[1]

[1] Département de Physique, de Génie Physique, et d'Optique, Université Laval, Québec (Québec), Canada G1V 0A6
May 13, 2018

We used deep convolutional networks to estimate the camera settings, lens aperture, ISO sensibility and exposure time, known as the EXIF metadata solely based on the pixels of the photos. The training has been performed on a novel dataset of 19 000 high quality photos labeled with the camera settings. The results indicated that deep convolutionnal networks have trouble solving the task and acheiving high accuracy. We find that the general low performances we obtained are due to the dataset corruption, some class imbalances and possibly the lack of information in the picture alone.

1 Introduction

Deep learning applications have recently provided important advances in the field of computer vision [30, 23, 13]. Indeed, using deep convolutional neural networks (CNN) in the context of features learning, these applications have led to groundbreaking results in image label prediction [28, 29, 15, 27], object detection [11], and object tracking [31] among others. These significant performance improvements were made possible due to the increasing training efficiency of these CNN architectures [12, 13].

While these applications focus primarily on the content of images, others have addressed problems related to their form. Examples of these would be image-style transfer [10], image resolution increasing [6, 7], and image orientation [26]. Even though these tasks focus on the form of an image rather than its content, they can still take full advantage of the feature extraction provided by these CNN architectures [15]. Among them belongs the learning of the exchangeable image file (EXIF) metadata of an image [9, 14], which characterizes the camera settings. The EXIF metadata has been used in the context of classical machine learning to improve a variety of tasks such as image classification [3, 4], image enhancement [24], image authentification [22], detection of people composite on pictures [21] and image search [17, 16].

Recently, a deep CNN framework has been proposed for learning the EXIF metadata from pictures [14]. Their objective was to extract three EXIF settings, namely the lens aperture, the photographic film's sensibility and the exposure time, solely based on the pixels of an image from the labeled dataset MIRFLICKR-1M [19]. Their strategy was to, first, extract a content-based feature foundation from a conventional CNN architecture, namely a pretrained VGG-16 on ImageNet [25] which was kept frozen during the training. These features were then combined using additional convolutional layers followed by fully connected layers trained in a supervised way to estimate the three EXIF settings.

Instead of predicting the three metadata using a single model, they trained three models, each predicting a distinct EXIF setting. To ease the problem, rather than estimating the parameters themselves, their model predicted for each EXIF setting one of two classes, low and high setting values, in which it belongs to. This problem turned out to be a difficult one, even in the simple case of binary classification. They showed that, while their model performed 79.44% on predicting the exposure time, their results on the other two remained inconclusive. They discussed that the sensibility and the aperture could be fundamentally more difficult to estimate because of the lighting conditions and the architecture they used.

In this paper, we investigate the similar problem of estimating these three EXIF settings in the context of multi-label classification [5, 32, 2]. More specifically, we consider the EXIF settings to be labels individually dis-

tributed in multiple classes. Then, assuming that these labels are correlated [2], meaning that some EXIF configurations are more frequent than others, the task is to evaluate the EXIF settings all at once with a single deep model. This task is performed on a novel dataset of 19 000 labeled images mined from the `unsplash.com` database [1].

The paper is organized as follows: In Sec. 2, our general method is exposed in detail. Namely, the dataset we used to train our CNN model, the model itself and the training techniques we considered for this problem are shown. Then, in Sec. 3, different training scenarios are investigated. Among other things, we determine which of the model's configuration was performing the best on the test dataset. Section 4 is dedicated to a discussion on the general results and the performances we obtained from the best model. We identify some issues with the problem itself and the dataset. Finally, in Sec. 5, we conclude with a brief overview of the paper's main contributions and some related future improvements.

2 Methods

2.1 EXIF Metadata

Exchangeable image file format (EXIF) metadata contains the information about how a photo was taken. Apart from the geolocalisation and the camera brand, we usually use EXIF to save the three main camera settings: the lens aperture, the ISO sensibility and the exposure time. These three settings control most of the fundamental aspects of photos.

The aperture, denoted x_{aper}, is the size of the opening from which the light enters into the camera. The larger is the aperture, the more light we can collect which results in a brighter image. Additionally, the aperture controls the depth of field of the image, i.e., the distance where objects appear in focus. Smaller depth of field is achieved with a low aperture. For close portraits, photographers usually prefer to use high apertures to obtain an artistic blur behind the subject to emphasize solely on the subject face. For scenery photography, it is recommended to use a small aperture to collect a high resolution on a broad domain.

The ISO, denoted x_{iso}, is a measure of the sensibility of the photographic films. It has a direct impact on the luminosity of the images and consequently controls the level of noise. A low ISO, e.g., 100, is usually used for bright subjects and leads to smooth images. Contrariwise, high ISO, e.g., 32 000, is used for dark scenery and introduce a high level of noise. This noise appears in the form of grains in the picture and can sometimes be used in portraits as an artistic effect. Nowadays, most digital cameras are able to select ISO from 100 to 64 000, and some specialized equipment can reach up to 4 000 000.

The exposure time, denoted x_{expo}, corresponds to how long the photographic films collect the light. Similarly to the ISO and the aperture, it has a major impact on the luminosity of the image. The exposure time must also be set depending on the velocity of the subject. Too long a exposure time when shooting a fast-moving object might result in high blurriness which can be, or not, desired.

Some combinations of $(x_{\text{aper}}, x_{\text{iso}}, x_{\text{expo}})$ are more frequent than others. For instance, in order to photograph a landscape, it is common to use low ISO, small aperture and high exposure time to capture fine static details at different distances. In contrast, shooting portraits usually involves high aperture and low exposure time to focus solely on the person's face. Therefore, the values of the three EXIF settings are expected to be correlated with each other depending on the type of picture.

2.2 Unsplash Dataset

Most curated image datasets do not contain the EXIF metadata required for this problem. The MIRFLICKR-1M dataset [19] is an exception where 5406 images have been reported to contain usable EXIF metadata [14]. Unfortunately, we consider that this dataset was too small for the task at hand. Indeed, splitting the dataset into training, validation and test subsets would leave too few images for training.

Thus, we collected 19 000 colored photos from `unsplash.com`, a website where users can publish high-quality photos, to form a new dataset (see Appendix 6 for detailed information on the image acquisition). As most images have 4k resolution, we resized them to 400×400 pixels and saved the aperture, the ISO and exposure time. Also, most images are in a 3:2 format, so we can safely assume that the resizing to a 1:1 format will not alter the image. We then randomly split the dataset into two sets:

Table 1: Bins intervals for each metadata for $N_{\texttt{bin}} = 3$.

	Bin		
	1	**2**	**3**
Aperture	$[0, 2.8[$	$[2.8, 4.9]$	$]4.9, [$
ISO	$[0, 215[$	$[215, 464]$	$]464, [$
Exposure time	$[0, 0.002[$	$[0.002, 0.046]$	$]0.046, [$

one of 5 000 images for test and 14 000 images for training and validation. For each training, 2 000 images were used for validation.

2.3 Binning

Instead of inferring the exact values of the EXIF metadata, we assigned metadata to bins and learned to infer the bins the photo metadata belongs to [14]. We separated each metadata into $N_{\texttt{bin}}$ groups of size c (see Tab. 1). In each case, we used linear bins for the aperture and logarithmic bins to group the ISO and the exposure time. In the logarithmic case, we set the lower bound $x_{\texttt{min}}$ and the upper bound $x_{\texttt{max}}$ and fix c so that

$$x_{\texttt{min}} = x_{\texttt{max}} c^{N_{\texttt{bin}}}. \tag{1}$$

Then, the first bin contains all the elements between 0 and $cx_{\texttt{min}}$, the second bin contains all the elements between $cx_{\texttt{min}}$ and $c^2 x_{\texttt{min}}$, and so on until the last bin contains the elements larger than $x_{\texttt{min}} c^{N_{\texttt{bin}}-1}$.

For linear bin sizes, we used

$$c = (x_{\texttt{max}} - x_{\texttt{min}})/N_{\texttt{bin}}. \tag{2}$$

Then, the first bin contains the elements between 0 and $x_{\texttt{min}} + c$, the second bin contains the elements between $x_{\texttt{min}} + c$ and $x_{\texttt{min}} + 2c$, and so on until the last bin contains the elements larger than $x_{\texttt{min}} + c(N_{\texttt{bin}} - 1)$. In all the following results of Sec. 3, we used $x_{\texttt{min}} = 0.7$ and $x_{\texttt{max}} = 7$ for the aperture, $x_{\texttt{min}} = 100$ and $x_{\texttt{max}} = 1000$ for ISO and $x_{\texttt{min}} = 1 \times 10^{-4}$ and $x_{\texttt{max}} = 1$ for the exposure time.

2.4 Oversampling

Some camera settings seem to be more common than others and we found that the dataset is suffering from severe class imbalances. The most eloquent example is the fact that about 69% of the images have the same value of 1/4000 seconds for the exposure time. Without a special sampling procedure, the network could learn the class imbalances and could fail to pick up the relevant features.

The most common methods to smooth the class imbalances are oversampling and undersampling [20]. While the latter is more efficient to sample an imbalanced dataset correctly [8], it is impractical in our case since it would significantly reduce the training dataset size. Part of the problem is due to the fact that settings belong to three types, i.e. aperture, ISO sensibility, and exposure time. Hence, we are ultimately sampling images the different combinations of these labels and are limited by the least frequent combination.

Instead, we used oversampling on the training dataset. First, we classify the image metadata into bins according to their values. For instance, a photo which has its aperture belonging to the third bin, its ISO to the first one and the exposure time to the second would be assigned the coordinate (2,0,1). Finally, we randomly pick a coordinate from that three-dimensional space and sample an image at this random position. In doing so, we uniformly sample images for all the metadata combinations.

We point out that this method has been reported to slightly increase overfit and bring additional noise to the training [20]. While this is unfortunate, oversampling was necessary during the training phase, because we found that without it the model would learn to always predict the most frequent settings without generalizing the problem.

2.5 Neural Network Model

The three EXIF parameters $(x_{\text{iso}}, x_{\text{aper}}, x_{\text{expo}})$ are usually correlated with each other depending on the type of pictures. This correlation implies that one should not divide the three associated learning tasks, for instance by training three separate neural networks, in order to maximize the final outcome performance [2, 14]. Instead, we consider a single multitask neural network model whose purpose is to determine those three all at once.

The architecture of the neural network model is divided into two main components: the image model and the tail model (Fig. 1).

Image Model

The role of the image model is to extract features related to the form of a picture, for instance blurriness, brightness, high resolution, etc. These are learned by standard CNN architectures from image recognition and classification

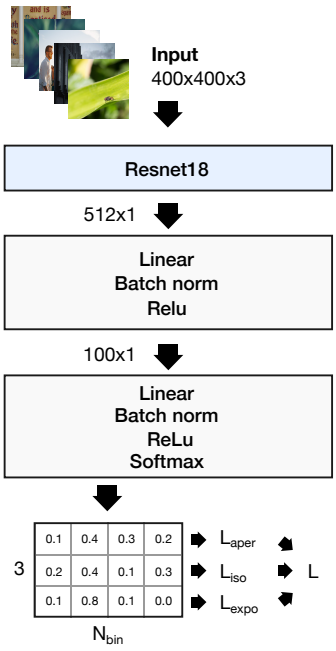

Figure 1: Schematization of the architecture.

[25]. More specifically, we considered two versions of ResNet [15]: ResNet-18 and SE-ResNet-18 with pretrained configurations on ImageNet [18]. Because we are not interested in the classification power of these CNNs, all the parameters are updated during the training phase.

Moreover, the fully connected layer usually at the end of this architecture is removed. The image model instead returns a vector given by the global average pooling of its last layer representing the average strength of each feature map.

Tail Model

The features extracted from the image are combined in the tail model. This model is composed of 2 fully connected hidden layers of 100 neurons, that each contains a linear transformation, a batch norm and a ReLU activation function (Fig. 1).

The tail model returns a vector of length $3N_{\texttt{bin}}$. The vector is resized into a matrix of size $3 \times N_{\texttt{bin}}$ and we apply a softmax function to each column. For instance, the ith element of the jth column corresponds to the probability that an image has its EXIF setting j belonging to bin i.

2.6 Loss Function

The model learns multiple labels at once. Therefore, the loss function L is divided into three main components, namely

$$L = \alpha_{\text{aper}} L_{\text{aper}} + \alpha_{\text{iso}} L_{\text{iso}} + \alpha_{\text{expo}} L_{\text{expo}} \tag{3}$$

where L_i is the loss evaluated on EXIF setting i and each term of the loss function is weighted with α_i. These weights allow us to tune the rate at which a given EXIF setting is learned. For instance, if $\alpha_{\text{aper}} = 1$ and $\alpha_{\text{iso}} = \alpha_{\text{expo}} = 0$, the model would only learn x_{aper} because no contribution of x_{iso} and x_{expo} would not contribute to the global loss L.

Learning each EXIF setting separately is understandably an easier learning task than learning the three at once, while it can lead to lower performances. As a learning strategy, we exploit this principle by letting the weights α_i be time-dependent. More specifically, the three weights α_i are linearly increasing functions of the current number of epochs T and are fixed by the following expressions

$$\alpha_1 = 1\,; \tag{4a}$$

$$\alpha_2 = \begin{cases} 0, & \text{if } T < \tau \\ \min\{1, \frac{T}{\tau} - 1\}, & \text{if } \tau \leq T < 2\tau \end{cases}; \tag{4b}$$

$$\alpha_3 = \begin{cases} 0, & \text{if } T < 2\tau \\ \min\{1, \frac{T}{\tau} - 2\}, & \text{if } 2\tau \leq T \end{cases}; \tag{4c}$$

where τ is the number of transitioning epochs and fixes the loss schedule. While the order in which the EXIF settings are sequentially learned is arbitrary, we suppose that learning the most difficult one earlier optimizes the learning rate of the model. For these reasons, we fix the first setting to be learned as the aperture x_{aper} followed by the ISO x_{iso} and then the exposure time x_{expo}.

In regard to the loss functions themselves, we used the standard cross-entropy loss

$$L_i \equiv L(\hat{\boldsymbol{y}}_i; \boldsymbol{y}_i) = \sum_j y_i^j \log \hat{y}_i^j \tag{5}$$

where $\hat{\boldsymbol{y}}_i$ is the softmax output vector of the model for a given EXIF setting i and $\boldsymbol{y}_i$ is the target one-hot vector.

2.7 Training details

To train the model, we use a stochastic gradient descent with Nesterov momentum factor $\beta = 0.9$. We also use a L2 penalty on the weights with a weight decay $\omega = 10^{-3}$ on every training and a learning rate schedule which reduces by a multiplicative factor of 0.1 when the validation

loss increases for 3 consecutive epochs. Unless specified, we used a batch size of 32, a number of bins $N_{\texttt{bin}} = 3$ and a number of transitioning epochs of $\tau = 20$. The maximal number of epochs is set to $T = 100$ but most training phases are completed after 60-70 epochs. For data augmentation, we added 10 pixels of zero padding to the resized images and a random rotation between $-120°$ and $120°$. For each training of T epochs, we only conserve the configurations which have performed the lowest loss on the validation dataset.

3 Experiments

In this section, we perform some experiments to find the most efficient model in this learning problem. In general, tuning the hyperparameters, we find that $\tau = 5$ and $N_{\texttt{bin}} = 3$ yield an adequate compromise between accuracy and generalization. Additionally, we find that SE-ResNet-18 was the most efficient architecture out of the models we considered.

3.1 Effect of Loss Schedule

The loss schedule controls the weights α_i of the losses L_i at each epoch and the hyperparameter τ controls how slow each L_i contributes to the global loss L. If $\tau = 0$, then all losses contribute during the whole training to the global loss. Here, we investigate the impact of this hyperparameter (Table 2). The worst accuracy is achieved when $\tau = 0$, indicating that the loss schedule improves and eases the learning. While we observe that $\tau = 5$ achieve the lowest loss, the case $\tau = 10$ has a higher accuracy on the aperture and the ISO. Furthermore, we find that when τ is too large, the aperture overfits before the ISO and exposure time are properly learned. Thus, since the aperture overfitting increases the loss, the best score on the validation dataset occurs early in the training process and we obtain poor accuracy on the ISO and exposure time. For these reasons, we find that $\tau = 5$ is a good compromise between learning gradually the problem and not overfitting too early.

3.2 Effect of training dataset size

We investigate the impact of the training dataset size on the accuracy (see Tab. 3). With fewer than 500 images for training, the accuracy is lower than a random draw, indicating that the network overfits the training dataset. We can also conclude that 5000 images seem to be the minimum to extract relevant features and generalize properly.

Table 2: Accuracy and losses on the test dataset for different periods of the loss schedule.

	Loss schedule period			
	0	**5**	**10**	**20**
Test loss	3.0297	**2.9868**	3.0039	3.1161
Aperture	39.98	42.41	**43.35**	41.95
ISO	48.26	50.19	**50.99**	50.67
Exposure time	53.87	**63.19**	60.66	57.56

Table 3: Accuracy and losses on the test dataset for different training dataset sizes. The baseline accuracy for random draw is 0.33.

	Training dataset size ($\times 10^3$)				
	0.1	**0.5**	**1**	**5**	**12**
Test loss	4.105	3.540	3.491	**3.176**	3.284
Aperture	31.23	37.53	38.31	34.59	**41.95**
ISO	25.42	40.09	44.17	**51.53**	50.67
Exposure time	17.36	46.67	54.50	**58.76**	57.56

3.3 Effect of the number of bins

Metadata are grouped into bins to simplify the inference. For instance, two bins reduced the problem to a binary classification: high or low camera settings. Thus, reducing the number of bins also reduce the difficulty of the problem. We examined the impact of the number of bins on the accuracy by training SE-ResNet-18 for different numbers of bins (see Tab. 4). As expected, the more bins we have and the lower is the accuracy. An interesting measure is the normalized average accuracy of the trained networks in comparison with the random draw. Indeed, it is increases with the number of bins. For instance, the network with 10 bins is on average 2.43 times as good as the random draw while the network with 2 bins is only 1.28 times better.

We find that the best compromising value for the number of bins is $N_{\texttt{bin}} = 3$. Indeed, while it performs on average much better than the random draw by a factor

Table 4: Accuracy of the inference for different numbers of bins on the test dataset. The probability of drawing one of the bins at random is indicated as a baseline. The normalized average accuracy (NAA) is the average accuracy of the model divided by the random draw accuracy.

	Number of bins			
	2	**3**	**5**	**10**
Random	50.00	33.33	20.00	10.00
NAA	1.28	1.50	1.59	2.43
Aperture	61.52	41.95	25.06	14.05
ISO	61.27	50.67	31.44	21.46
Exposure time	69.86	57.56	39.25	34.36

Table 5: Accuracy of the inference for different CNN architectures.

	ResNet-18	**SE-ResNet-18**
Aperture	41.91	**41.95**
ISO	**51.43**	50.67
Exposure time	49.5	**57.56**

1.5, the accuracy remains relatively high in that configuration.

3.4 Effect of architecture

The quality of the inference is grounded into the quality of the features extracted from the images. Hence, we considered different versions of ResNet on our learning task: ResNet-18 from Ref. [15] and SE-ResNet-18 from Ref. [18]. Using different architectures will extract different types of features and it will indicate if feature quality is important for the inference.

We compared the impact of adding SENet modules between each ResNet blocks. The role of SENet modules is first to gather the features at a given layer and second to recalibrate them. In a sense, the SENet module is a form of attention on the features. We therefore expect the SE modules to increase the specificity of the extracted features. We found no significant improvement for the aperture and ISO accuracy but an increase of 8% for the exposure time accuracy.

4 Discussion

In the previous section, we have exposed a variety of techniques to tackle the problem of estimating the camera settings from the pixels of photos. In general, we found that this problem was difficult to address while we learned that it was somewhat possible to extract some information. In this section, we discuss the quality of the deep learning algorithm to extract the EXIF settings based the previous results.

4.1 Best and Worst Examples

To evaluate the performance of the model, we looked at the best and worst examples of the test dataset predicted by our best model. We ranked the performance of the model according to the loss value of the images. In Fig. 2, we show the top 5 of the best and worst examples and indicate the prediction of each EXIF settings.

At first glance, we notice that all the best examples have high x_{iso} and x_{expo} while the worst examples have them distributed between low and high values exclusively. This could mean that high exposure times and high ISO images are easier to process. The fact that high ISO and high exposure times are related through bright images might explain this. Thus, the network may have detected that bright images fail under the same category.

When the background is well defined and distant from the central objects, the inference seems easier. In contrast, for the top 1 worst example, there is no background and the central object is spread along the whole picture. Hence, even for the human eye, it is rather difficult to detect any depth in the picture and therefore to give any value to x_{aper}. This lack of identifiable depth is present in all the considered worst examples except for the fifth case.

In general, for the best scenarios, the model is quite confident in its predictions. That is, the bin distribution is often peaked around a single value of setting corresponding to the ground truth. On the contrary, for the worst scenario, the bin probabilities are often close to evenly distributed.

4.2 Label Correlations

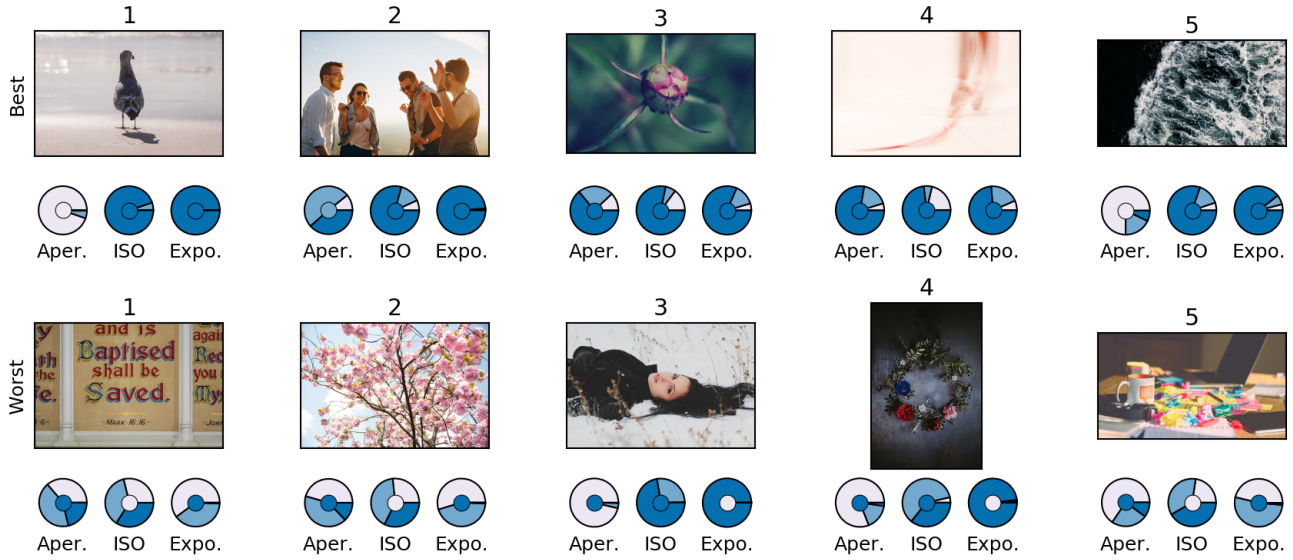

Figure 2: Display of the top 5 best (top) and top 5 worst (bottom) examples predicted by the model in the test dataset. Underneath each picture is shown in the form of pie charts the probability distribution of the three bins of each setting given by the model. Light blue indicates low setting, medium blue for medium setting and dark blue for high setting. The largest portion of the chart corresponds to the predictions and the colors of the center display the targeted settings.

We investigated the impact of the output format of the last layer. Two formats were investigated: (i) the independent or (ii) the correlated inference. For the former, the network output is composed of three rows of length equal to $N_{\texttt{bin}}$ and individually normalized so that each row represents the probability distribution of the metadata, i.e. $P(x_{\text{iso}})$ or $P(x_{\text{aper}})$ or $P(x_{\text{expo}})$. It is the typical output described at Sec. 2.5. The joint probability is measured assuming the independence of their prediction:

$$P(x_{\text{iso}}, x_{\text{aper}}, x_{\text{expo}}) = P(x_{\text{iso}})P(x_{\text{aper}})P(x_{\text{expo}}) \qquad (6)$$

The correlated format outputs a single probability vector which elements give the joint probability $P(x_{\text{iso}}, x_{\text{aper}}, x_{\text{expo}})$ of observing a single configuration of the metadata. In this case, the length output is equal to the $3^{N_{\text{bin}}}$. In this case, a single cross-entropy loss function is used without loss schedule.

Table 6 shows the accuracy of the model to correctly predict different numbers of settings for these two output formats. We find that the independent output has a higher probability to detect more metadata on an image than the correlated output. On average, the independent output obtain 1.48 correct metadata in comparison to 1.27 for the correlated output and 1.0 for the random draw.

Table 6: Accuracy to infer different number of metadata for two types of output. The random draw accuracy is indicated as a baseline.

	Number of metadata			
	0	**1**	**2**	**3**
Random draw	29.62	44.44	22.22	3.70
Independent	12.2	38.2	38.04	11.56
Correlated	20.94	39.52	30.26	9.28

Indeed, since the correlated output delineates a much larger number of classes, it is understandably a harder problem to solve. However, the gain in performance for the independent format is not significantly large. In general, these two representations seems quite equivalent and are both better than the random draw baseline.

4.3 Importance of characteristic dimensions

It is quite clear that the model has been able to extract some features. Indeed, as seen in Tabs. 3 and 4, the scores achieved on the test dataset are above that of the random draw baselines. This means that the model is indeed able to classify in part the images based on some extracted

features and generalize.

However, the overall performances are undeniably poor. There are some intuitive reasons why this is so and one of them is related to the fundamental difficulty of the problem. Indeed, the EXIF settings are intrinsically related to at least three dimensions which are hard to extract from the pictures alone: the notions of length, brightness and motion of a picture's surrounding environment.

First, it is certainly hard for the model to distinguish objects that are big and distant in reality. This problem is related to the characteristic lengths of an image. This is a major issue because it is often those characteristic lengths that let us distinguish the depth of field. Therefore, it is limiting our predictive power over x_{aper}. For instance, in Fig. 2, the flower appearing in the third-best example is surely close to the camera while the tree on the second-worst example is surely far from it. The fact that the flower appears to be of the same size as the tree is an artifact of the picture. This is not captured by the model because it was not taught that flowers and trees are generally quite different in size. This effect is maximized when the picture does not have different depth plans stacked on one another similarly to the first-worst example. While the characteristic lengths of a picture can be inferred based on the context and the objects displayed on it, similar to what humans do, it requires the model to learn the categories of these objects in an unsupervised manner. Thus, if we had taught the model to predict the categories of the images as well as the EXIF metadata, we would have expected a higher accuracy on the EXIF metadata.

The second characteristic dimension which makes the inference more difficult is the ambient brightness of a picture. Albeit it can be estimated for night scenery, the effect of the ISO setting on the brightness of a picture makes it hard to estimate in most cases. Hence, the relation between the brightness and x_{iso} can hardly be unveiled using the picture alone. The only information contained by the image is the amount of grain (or noise) present in the image which delineates low and high ISO. What we found is that it is the relationship of x_{iso} with the other two settings that makes its inference possible in most cases. Low aperture usually reduces the amount of light penetrating the camera, thus resulting in a need for higher ISO, for example.

Lastly, given that the main object of a picture is moving, a long exposure time will leave blurry traces in a picture. However, depending on the velocity of that object and the length of the exposure time, the final results can be quite similar. The lack of knowledge for the characteristic motion of a picture, also based on its context, is a problem to infer correctly the exposure time.

From this discussion, it is clear that extracting the EXIF settings from the pixels of a picture is a difficult problem in itself. In general, the model needs a bigger understanding of the picture's context in order to precisely estimate its camera settings. Detecting the objects present in the picture could improve the predictions.

4.4 Corruption of the Dataset

Datasets of images labeled with their EXIF settings are rare. In fact, the MIRFLICKR is to our knowledge the only one available and among its 1 000 000 labeled images, only about 5 000 are labeled with the camera settings. This justified our need to build the Unsplash dataset [1].

Among the labeled images we collected, we found some which contained bad EXIF settings for instance with $x_{\text{aper}} = x_{\text{iso}} = x_{\text{expo}} = 0$. When that was the case, those images were removed from the dataset. We also suspected, based on our photography expertise, that some notwithstanding errors of settings appeared throughout the dataset which is much more difficult to find. For example, images in dark environment where x_{iso} was abnormally low for that kind of scenery can be found in the dataset.

The pictures available in the Unsplash dataset are taken by enthusiastic and, perhaps, professional photographers. Therefore, most of the pictures are high quality, highly filtered and even numerically enhanced for artistic purposes. For inferring the camera settings, modifying pictures in such way can alter some of its features essential to the inference. Those manipulation introduce noise in a picture and they surely resulted in an additional increase of the learning task difficulty.

4.5 Bin Population Versus Bin Significance

The number of bins has been found to be an important aspect in the learning process (see Sec. 3.3). But the absolute position of the bins and their sizes are also critical hyperparameters. They fix how many images are in each bin and how much they relate to each other in terms of their settings. We were then exposed to a dilemma: bin population versus bin significance. For instance, the three

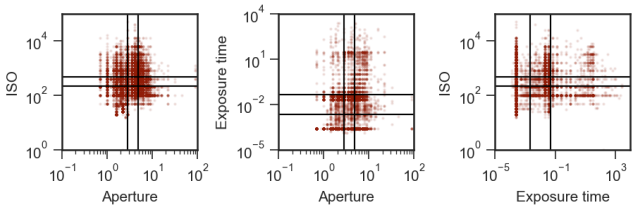

Figure 3: Distribution of examples in the setting space $(x_{\text{aper}}, x_{\text{iso}}, x_{\text{expo}})$. The axis of each plot use logarithmic scales. In each subset, a marker indicates the presence of at least one example in that configuration and its intensity indicates its frequency.

bins for ISO of intervals $[0, 125]$; $]125, 320[$; $[320, 30000]$ would split evenly the dataset. However, the luminosity different between ISO = 400 and ISO = 300 is extremely weak and excessively different compared to ISO = 16 000. Thus, a uniform distribution would induce inner-bin heterogeneity and weaken the contrasts between bins. It results in a poor inference due to the incapability to distinguish two bins and to associate relevant bin specific features.

On the other hand, uneven bin populations can occur when we manually set the intervals covering different spectrum of settings. The fact that we favored bin population over bin significance may also have contributed to the poor inference accuracy we ultimately obtained.

5 Conclusion

In this paper, we investigated the task of learning the camera settings, known as EXIF metadata, from the pixels of pictures using deep learning techniques. While we found that this task is a difficult one [14] even in the context of deep learning, we showed that some information regarding those settings can nonetheless be extracted.

In comparison with the results from Ref. [9], ours are certainly promising. We showed that label correlation, which was not considered in this work, was important in that problem. Indeed, we were able to correctly predict the aperture with 42.41% of accuracy and the ISO sensibility with 50.19% considering three bins. The only decrease in performance is related to the exposure time where our model predicted it with 57.56% of accuracy on the test dataset. However, we found that our performances remain rather poor in contrast with other similar tasks such as image recognition.

In general, two main issues prevented us from completing the task at hand. The first one is related to the problem itself: The pictures alone usually do not contain enough information to estimate accurately their EXIF settings. In fact, it is the main characteristic dimensions of a picture, practically essential for the inference, that are missing. The second one was the quality of the datasets which was crucial for the success of the inference. The imbalance of the classes in the dataset was partially solved by oversampling techniques and appropriate binning, but the bad labeling remained an issue.

There is a number of ways which could improve this experiment. First, it would be necessary to make a similar analysis with the MIRFLICKR-1M dataset used in Ref. [14]. While it was not discussed in their paper, we suspect that class imbalance was an issue in that dataset as well. Second, the fact that the images were reshaped to 400×400 pixels could be a problem. A patching technique could be used to reduce the dimension of a picture while preserving its resolution. Finally, we used small models to make the inference. While adding SENet modules turned out to increase the performance for the exposure time estimation, it did not change the performance of the other two settings. It would be interesting to investigate deeper models in order to verify if, by increasing the generalization power of the features, the performance increases as well.

6 Acknowledgements

The authors would like to thank Philippe Giguère for insightful discussions. This work was supported by the Fonds de recherche du Québec-Nature et technologies (EL), and the program Sentinel North, financed by the Canada First Research Excellence Fund (EL, CM). We thank Compute Canada and Calcul Québec for their infrastructure.

Appendix A: Dataset Acquisition

Dataset acquisition has been made possible using the `Unsplash API`. We used an entry point for which the server returns 30 random photos that match a specific query word. We use a set of 1000 popular words in English to send as queries. In doing so, the images match a large range of subjects, from the portrait, scenery and all kinds of objects. Only images with EXIF metadata was saved and duplicates have been removed. Out of 24 000 extracted images, 19 532 images contained the EXIF metadata. Figure 4 displays a sample of the complete dataset. We normalized the images using the measured averages and variances (Tab. 7).

Table 7: Averages and variances for RGB channels used for the dataset normalization.

	R	G	B
μ	0.38784351	0.37949627	0.364951
σ^2	0.26650685	0.25285939	0.2522641

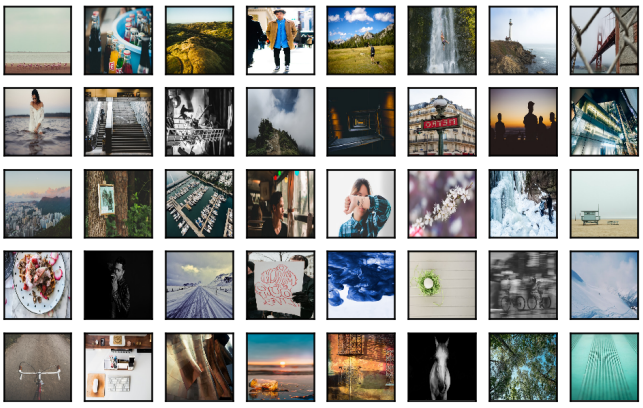

Figure 4: Images sampled from the Unsplash dataset.

References

[1] *Unsplash*. https://unsplash.com/, 2018. [Online; Accessed May 9th, 2018].

[2] W. Bi and J. T. Kwok, *Multilabel classification with label correlations and missing labels.*, in AAAI, 2014, pp. 1680–1686.

[3] M. Boutell and J. Luo, *Photo classification by integrating image content and camera metadata*, in ICPR, vol. 4, IEEE, 2004, pp. 901–904.

[4] ———, *Beyond pixels: Exploiting camera metadata for photo classification*, Pattern Recognit., 38 (2005), pp. 935–946.

[5] S. S. Bucak, R. Jin, and A. K. Jain, *Multi-label learning with incomplete class assignments*, in CVPR, 2011, pp. 2801–2808.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, *Learning a deep convolutional network for image super-resolution*, in ECCV 2014, 2014, pp. 184–199.

[7] ———, *Image super-resolution using deep convolutional networks*, IEEE Trans. Pattern Anal. Mach. Intell., 38 (2016), pp. 295–307.

[8] C. Drummond, R. C. Holte, et al., *C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling*, in Workshop on learning from imbalanced datasets II, vol. 11, Citeseer, 2003, pp. 1–8.

[9] J. Fan, H. Cao, and A. C. Kot, *Estimating EXIF parameters based on noise features for image manipulation detection*, IEEE Trans. Inf. Forensics Secur., 8 (2013), pp. 608–618.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, *Image style transfer using convolutional neural networks*, in CVPR, 2016, pp. 2414–2423.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, in CVPR, 2014, pp. 580–587.

[12] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, in AISTATS, 2010, pp. 249–256.

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1, MIT Press Cambridge, 2016.

[14] D. Gupta and S. Kannan, *EXIF estimation with convolutional neural networks*, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in CVPR, 2016, pp. 770–778.

[16] M. Hirota, N. Fukuta, S. Yokoyama, and H. Ishikawa, *A robust clustering method for missing metadata in image search results*, J. Inf. Process., 20 (2012), pp. 537–547.

[17] M. Hirota, S. Yokoyama, N. Fukuta, and H. Ishikawa, *Constraint-based clustering of image search results using photo metadata and low-level image features*, in Computer and Information Science 2010, Springer, 2010, pp. 165–178.

[18] J. Hu, L. Shen, and G. Sun, *Squeeze-and-excitation networks*, arXiv preprint arXiv:1709.01507, (2017).

[19] M. J. Huiskes and M. S. Lew, *The MIR flickr retrieval evaluation*, in MIR, 2008, pp. 39–43.

[20] N. Japkowicz and S. Stephen, *The class imbalance problem: A systematic study*, Intell. Data Anal., 6 (2002), pp. 429–449.

[21] M. K. Johnson and H. Farid, *Detecting photographic composites of people*, in IWDW 2007, Springer, 2007, pp. 19–33.

[22] E. Kee, M. K. Johnson, and H. Farid, *Digital image authentication from JPEG headers*, IEEE Trans. Inf. Forensics Secur., 6 (2011), pp. 1066–1075.

[23] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature, 521 (2015), p. 436.

[24] I. V. Safonov, I. V. Kurilin, M. N. Rychagov, and E. V. Tolstaya, *Image enhancement pipeline based on EXIF metadata*, in Adaptive Image Processing Algorithms for Printing, Springer, 2018, pp. 65–83.

[25] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556, (2014).

[26] K. Swami, P. P. Deshpande, G. Khandelwal, and A. Vijayvargiya, *Why my photos look sideways or upside down? detecting canonical orientation of images using convolutional neural networks*, in ICMEW 2017, July 2017, pp. 495–500.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, *Inception-v4, inception-resnet and the impact of residual connections on learning.*, in AAAI, vol. 4, 2017, p. 12.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al., *Going deeper with convolutions*, in CVPR, 2015.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, in CVPR, 2016, pp. 2818–2826.

[30] R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.

[31] N. Wang and D.-Y. Yeung, *Learning a deep compact image representation for visual tracking*, in Adv. Neural Inf. Process Syst., 2013, pp. 809–817.

[32] H.-F. Yu, P. Jain, P. Kar, and I. Dhillon, *Large-scale multi-label learning with missing labels*, in ICML, 2014, pp. 593–601.