17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Efficient transfer learning for multi-channel convolutional neural networks

Aloïs de La Comble
Rakuten Institute of Technology
92 rue Réaumur, 75002 Paris, France
alois.delacomble@rakuten.com

Ken Prepin
Rakuten Institute of Technology
92 rue Réaumur, 75002 Paris, France
ken.prepin@rakuten.com

Abstract

Although most convolutional neural networks architectures for computer vision are built to process RGB images, more and more applications complete this information with additional input channels coming from different sensors and data sources. The current techniques for training models on such data, generally leveraging transfer learning, do not take into account the imbalance between RGB channels and additional channels. If no specific strategy is adopted, additional channels are underfitted. We propose to apply channel-wise dropout to inputs to reduce channel underfitting and improve performances. This improvement of performances may be linked to how much new information is brought by additional channels. We propose a method to evaluate this complementarity between additional and RGB channels. We test our approach on three different datasets: a multispectral dataset, a multi-channel PDF dataset and an RGB-D dataset. We find out that results are conclusive on the first two while there is no significant improvement on the last one. In all cases, we observe that additional channels underfitting decreases. We show that this difference of efficiency is linked to complementary between RGB and additional channels.

1 Introduction

Multi-channel datasets using domain specific image channels in addition to RGB channels have become more and more common [3, 29, 31, 4, 21, 16, 27]. However, none of these datasets has a sample size comparable to common large RGB datasets [5, 13]. These large RGB datasets are the ones generally used to pre-train convolutional neural networks (CNN) before transferring the learnt parameters as initial weights for another task. Those weights will be fine-tuned on a target dataset for which data is usually scarce. This process is called transfer learning [30, 24, 32]. [20] show that the smaller the target dataset, the higher the performances increase using transfer learning. It suggests there is a potential in using transfer learning to improve performances on these datasets. Applying transfer learning from an RGB dataset to a multi-channels dataset is not trivial in the sense that additional channels (channels that are not RGB and added after pre-training) cannot benefit from the same training as the RGB channels if no large annotated dataset with the same additional channels exists. Working with multi-channel data has been studied by [4, 17, 14, 7] proposing different ways of merging features. However, none of these works focus on the imbalance between RGB and additional channels existing at the beginning of fine-tuning. We show that models trained on multi-channels datasets, when evaluated only on additional channels, perform worst than model trained only on additional channels: models already trained on RGB foster on RGB, and additional channels are under-exploited (underfitted).

We propose an approach for fine-tuning an RGB-pre-trained CNN on a multi-channel dataset while forcing the model to fit on additional channels. This approach, input channel dropout, consists in randomly dropping some of the input channels. Only the inputs are modified, so it can be plugged into any CNN architecture without changing its inner layers. It makes it very easy to implement. It is comparable to a data augmentation technique.

We measure the improvement of this training strategy on three datasets, two of which show significant improvement, and measure a reduction in additional channel underfitting for all datasets. We set up a metric to measure the complementarity of additional channels relatively to RGB channels for the learning task. We show that lower improvements on a dataset is due to weaker complementarity.

In the next section, we present the works related to this paper. We then present input channel dropout in details and then perform computational experiments on three object detection multi-channel datasets. We finally discuss the results and how input channel dropout improves performances by better exploiting additional channels.

2 Related Work

Three aspects of Neural Networks training are involved together when adding new channels to already existing network: the **fusion** of these new channels with the previous ones, the **transfer learning** between classic channels (RGB) and less common chan-

nels (depth, multi-spectral, pdf-metadata), and the **regularization** used when data is unbalanced.

Concerning **fusion**, two cases can be distinguished: the channels are from different modalities (image, sound, video, text) or the channels are from the same modality (image RGB, image Depth, image spectral). In the case of channels from different modalities, we would talk about *multi-modal fusion* where as in the case of the same modality, our case, we would talk about *multi-channel fusion*. Fusion can be done earlier or later, either in multi-modal context [2] or multi-channel context [6].
In our case, multi-channel fusion, the different channels being of same nature (similar dimension, similar way of processing them), they can be fused at earlier stage without requiring much modification of the architecture. That makes this choice relevant for testing the potential of a new channel without redesigning the whole model or when processing power is limited (we train only one model).

When using datasets with limited amount of data, models pre-trained on huge RGB data sets can be leveraged: previous works [4, 29, 17, 14, 7] use **transfer learning** from ImageNet [5] to multi-channel datasets (RGB + near infra-red, RGB + depth, mix of infra-red, motion and gray-scale images).
The pre-trained model is optimized for using RGB channels but no pre-training is done on other channels. Input channel dropout aims at limiting the underfitting of additional channels. Our method is close to **dropout** which forces the network to randomly disable neurons or connections [26], constraining features to have low co-adaptation. In CNNs, dropout [15] is most commonly used at the end of the network on fully connected layers. Adaptations are proposed for convolutional filters [28, 18, 8]. [25, 11] propose to drop a whole channel from the inner layers of a CNN at once. We propose to drop whole channels at the input level to force the model to learn from additional channels. In that case, the dropout can be interpreted on one hand as a way to ensemble multiple predictors like in feature bagging where multiple models are trained on subset of selected features before being aggregated to form a stronger predictor [22], and on the other hand as a way to augment data creating new samples from initial ones by removing some channels.
Concerning feature bagging, a noticeable remark from [22], is that the more complementary are the channels used for different predictors, the higher the improvement on the aggregated predictor. We thus propose a definition and an experimental measure of this complementarity between channels.
Concerning data augmentation, [10] make the distinction between conservative and aggressive data augmentation. Conservative data augmentation does not change the sample too much whereas aggressive data augmentation takes the risk to break some of the information present in the original sample.
When using input channel dropout, we can choose a drop rate similar to [15] that can control the aggressiveness of the regularization.

3 Methods

Code and datasets are available at `https://github.com/19327482/input_channel_dropout`

Channel underfitting measurement : For a model M and a set of channels F, we note $Score[M(F)]$ the performances of M evaluated on F only (replacing the other channels ($\bar{F}$) by zeros). We estimate the underfitting of a set of channels F (additional or RGB) for a model M by:

$$underfitting(M, F) = \frac{Score[M_F(F)] - Score[M(F)]}{Score[M_F(F)]}$$

we measure how far the performances $Score[M(F)]$ are from the maximal performances the network architecture can achieve on F. We estimate this maximum by training a model M_F on F and evaluating it on F: $Score[M_F(F)]$. We perform 3 trainings and evaluations for M_F and take the average mAP as the reference value.

Input channel dropout : We consider the case of training a CNN with multiple input channels including RGB using transfer learning. When using transfer learning, weights of the first channel featuring RGB channels are already able to extract relevant features. Weights of additional channels however were not trained. We choose to leverage RGB pre-training similarly as [4] : we initialize additional channels first layer weights using for each channel the average of the weights of the RGB channels.

We propose two variants of input channel dropout: independent and simultaneous drop. We call "independent" drop the standard setting where each channel is dropped independently from the others according to a Bernoulli distribution with parameter p_{drop}. We propose another variant taking into account the fact that RGB channels have benefited from pre-training. We drop RGB channels simultaneously instead of independently, increasing the probability that none of the RGB channels is kept. We call "simultaneous drop" the setting in which RGB channels are dropped or kept together and additional channels are dropped independently.

In both variants, to drop a channel, we replace its pixels values with zeros. If all channels are to be dropped, we instead keep them all in order to avoid adding noise to the training procedure. To keep the input values for the network consistent, we multiply the intensity of each kept channel by $n_{channels}/n_{kept}$.

Channel complementarity measurement: For each dataset, we estimate the complementarity of the set of additional channels to the RGB channels to perform the task (see figure 1). We measure the number

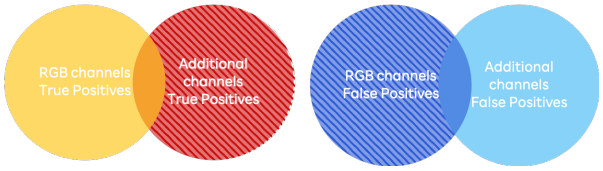

Figure 1. Venn diagram explaining the additional channel complementarity measurement. Striped part is the part we retain for counting.

of samples for which additional channels can correct the prediction of RGB channels. For this, we train and evaluate a model M_{add} using only additional channels and we train and evaluate another model M_{RGB} using only RGB channels. For each confidence threshold from 0.1 to 0.9 with a step of 0.1, we count all true and false positives of each model. We count the percentage $\Delta_{TruePos}$ of M_{add} true positives not included in M_{RGB} true positives (striped red part in 1) as the number of samples on which additional channels can bring recall improvement:

$$\Delta_{TruePos} = \frac{|TruePos(M_{add}) \not\subset TruePos(M_{RGB})|}{|TruePos(M_{RGB})|}$$

In the same way, we count the percentage $\Delta_{FalsePos}$ of M_{RGB} false positives not included in M_{add} false positives (striped blue part in 1) as the number of samples on which additional channels can bring precision improvement:

$$\Delta_{FalsePos} = \frac{|FalsePos(M_{RGB}) \not\subset FalsePos(M_{add})|}{|FalsePos(M_{RGB})|}$$

To assess if two predictions match, we compute their intersection over union and set a threshold at 0.5. We finally compute Δ_{add}: the harmonic mean of $\Delta_{TruePos}$ and $\Delta_{FalsePos}$ (as in F1-score computation) as a synthetic metric of additional channel interest for the task.

We train three M_{RGB} and three $M_{FalsePos}$ models and do the counting for each of the possible 9 couples.

4 Experiments

We run a set of experiments on three multi-channel datasets to measure the effect of input channel dropout. All of these datasets have object-level annotations. We use TensorFlow [1] object detection library's [12] implementation of Faster-RCNN [19] with a 50 layers ResNet [9] backbone. For each task, we measure the mean average precision (mAP) as defined in COCO challenge for different drop rates. Each model is pre-trained on COCO [13] dataset, weights are available at Tensorflow model zoo. We run each model 3 times and measure mean and standard deviation for mAP.

For this first experiment, we use the publicly available dataset "Multispectral Object Detection for Autonomous Vehicles" [23], composed of multispectral images (RGB, far-infrared, middle-infrared and near-infrared), with annotated [person, car, bike, color cone, car stop, bump, hole, animal]. This dataset is interesting for testing our approach because additional channels add crucial information for the task: some objects are much easier to recognize using certain channels, the complementarity between channels is high (see figure 2).

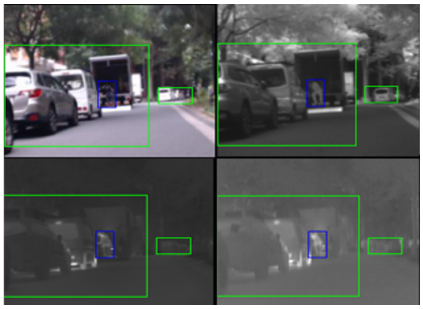

Figure 2. Top-left to bottom-right: RGB, near, middle, far infra-red. The person in the truck is most distinguishable on infra-red images. Cars on the other hand are more visible on RGB.

For the second experiment, we used a proprietary dataset composed of 7 magazines (424 pages). Magazine pages are labelled with bounding boxes around each paragraph of text and each illustration. To each bounding box is assigned a category: [pretitle, title, subtitle, column, illustration, caption, title2, title3, frame]. We used PDFalto to parse PDF files and extract bounding boxes and font-size for each line of text. We add three channels that spatially map font-related features in grey levels, pixel value in [0,255] (Fig. 3). The first channel is the font-size relative to the maximum font-size in the magazine. For each line of text, we draw a grey bounding box: the smaller the font-size, the darker the box. This feature contains some magazine-wise information not available in the RGB channels. The second channel is the font-size frequency in the magazine. The third channel is a paragraph and illustration indicator channel: close lines of text are grouped together in a grey (127) rectangle, paragraph bounding box; white (255) rectangle, bounding boxes of illustrations, are added when they could be parsed from the pdf.

The third dataset, is the RGBD EPFL-corridor dataset. It is composed of 6 scenes of up to 8 persons walking in a corridor. We used the corrected labels from [17]. We chose to split the data for training and testing scene-wise.

5 Results and discussion

Underfitting. We observe that in all cases, input channel dropout reduces underfitting of additional

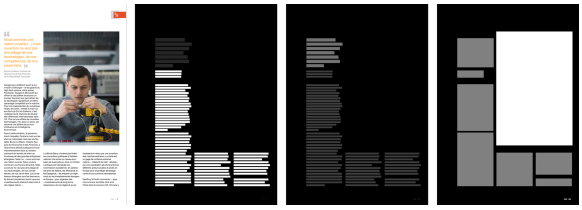

Figure 3. Left to right: RGB channels of the page, font-size frequency feature, relative font-size feature, paragraph / image indicator feature.

	Independent drop		Simultaneous drop	
Drop rate	mAP (%)	Underfitting RGB - Additional	mAP (%)	Underfitting RGB - Additional
	Multispectral dataset			
No drop	21.53 ± 0.80	37.4% - 57.5%	21.53 ± 0.80	37.4% - 57.5%
5%	23.04 ± 0.56	21.8% - 37.5%	22.72 ± 0.97	23.1% - 23.2%
10%	22.75 ± 0.31	17.6% - 35.7%	24.25 ± 0.23	15.6% - 14.0%
20%	23.83 ± 0.71	14.9% - 29.0%	23.71 ± 0.22	9.4% - 3.8%
30%	23.51 ± 0.46	9.7% - 20.8%	**24.43** ± 0.59	11.5% - 0.3%
40%	**23.89** ± 0.60	1.5% - 14.5%	23.42 ± 0.45	7.4% - 5.2%
50%	23.00 ± 0.60	5.2% - 10.3%	23.35 ± 0.26	0.2% - 1.5%
	Layout analysis dataset			
No drop	59.83 ± 0.57	27.4% - 80.4%	59.83 ± 0.57	27.4% - 80.4%
5%	**62.26** ± 1.08	4.5% - 57.4%	**62.18** ± 0.70	4.2% - 5.0%
10%	61.71 ± 0.36	2.6% - 40.6%	62.09 ± 0.47	3.1% - 6.3%
20%	61.71 ± 1.24	2.6% - 26.4%	61.16 ± 1.00	4.0% - 3.2%
30%	61.38 ± 0.51	3.2% - 13.8%	61.2 ± 0.34	0.2% - 0.4%
	RGBD pedestrian detection dataset			
No drop	64.54 ± 0.05	1.3% - 44.0%	**64.54** ± 0.05	1.3% - 44.0%
5%	**64.63** ± 0.47	2.0% - 33.2%	64.26 ± 0.53	10.1% - 5.2%
10%	64.54 ± 0.66	1.9% - 23.9%	64.04 ± 0.49	12.2% - 5.8%
20%	63.98 ± 0.09	4.7% - 15.5%	63.17 ± 0.14	2.7% - 5.2%
30%	63.52 ± 0.36	11.9% - 9.2%	64.05 ± 0.52	9.2% - 0.4%

Table 1. Results: mAP (average and standard deviation) and underfitting for all 3 datasets. No drop is the baseline. Input channel dropout reduces additional channels underfitting on all datasets. It improves performances on multispectral and PDF datasets.

channels (80.4% down to 0.4% for layout analysis, 57.5% to 0.3% for multispectral and 44.0% to 0.4% for RGBD with simultaneous drop 1). Input channel dropout also reduces underfitting for RGB channels for layout analysis and multispectral (resp. 27.4% down to 0.2% and 37.4% down to 0.2% with simultaneous drop) but not for RGBD dataset for which baseline underfitting of the RGB channels was already very low (1.3%). We observe that additional channels underfitting for baseline is always higher than RGB underfitting (at least $\times 1.5$), which confirms the imbalance between the two sets of channels that we attribute to pre-training.

Overall improvement. We observe (table 1) that input channel dropout improves mAP on multispectral (rel. +13.5%) and layout analysis (+4.1%) datasets for both drop variants. However this is not the case for the RGBD dataset (+0.1%).

Channels complementarity. $\Delta_{additional}$ estimates how much additional channels are complementary to RGB. We measured a value of 50.8% for multispectral dataset, 25.0% for layout analysis, and 6.5% for RGBD dataset. It appears correlated with the improvements brought by input channel dropout. We conclude that additional channels must be complementary to RGB channels to see an improvement using input channel dropout.

dataset	$\Delta_{TruePos}$	$\Delta_{FalsePos}$	$\Delta_{additional}$	RGB Underf.	Impr.	Best drop r.
Multisp.	36.1%	85.6%	50.8%	20.2%	13.5%	30%
PDF	15.3%	68.5%	25.0%	27.4%	4.1%	5%
RGBD	3.3%	61.0%	6.5%	1.3%	0.0%	No drop

Table 2. Additional channel complementarity measures for each dataset, intial RGB channels underfitting rate, simultaneous drop best relative improvement and best drop rate.

We also observe for the RGBD dataset that initial RGB channels underfitting is very low compared to other datasets and increases with the drop rate. We conclude that input channel dropout forces the model to learn on additional channels and that therefore it learns less efficiently on RGB channels.

Drop rate. The optimal drop rate for mAP depends on the drop variant, but also on the dataset: it is the highest for the multispectral dataset (30% for simultaneous, 40% for independent), it is 5% for the two variants for layout analysis dataset, and "no drop" gets the best performances for the RGB-D dataset. The optimal drop rate is correlated to the relative importance of additional channels in the dataset.

The higher the drop rate, the more the model is forced to learn from every channel: it reduces underfitting. However optimal drop rate is moderate and it does not minimize underfitting. Indeed, channel underfitting does not take into account features co-adaptation that might be beneficial to perform the task. There is a trade-off between fully exploiting additional channels and encouraging the model to learn to extract features from joint modalities.

Drop mode. Finally, we observe that simultaneous drop is able to further reduce additional channels underfitting using high drop rates. It also offers a lesser number of different input combinations to the model : at a given drop rate, data augmentation is less aggressive. This might explain the superiority of simultaneous drop on the multispectral dataset : it is able to better reduce additional channels underfitting while not adding too much aggressive perturbation to the input.

6 Conclusion

We have proposed a strategy to improve transfer learning from RGB to multi-channel datasets. It can be easily applied to any network architecture as data augmentation. We measured on three different datasets the improvement brought by our method for different hyper parameters choices. It appears that the more complementary are the additional channels relatively to RGB, the higher will be the improvement and the higher the optimal drop rate. This is due to a trade-off between reducing additional channels underfitting and

adding aggressive perturbations to the training that can also impede joint learning of features.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] T. Baltrusaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443, Feb. 2019.

[3] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3, Sep 2015.

[4] G. Choe, S. Kim, S. Im, J. Lee, S. G. Narasimhan, and I. S. Kweon. Ranus: Rgb and nir urban scene dataset for deep scene parsing. *IEEE Robotics and Automation Letters*, 3(3):1808–1815, 2018.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller, and W. Burgard. Multimodal deep learning for robust RGB-D object recognition. *CoRR*, abs/1507.06821, 2015.

[7] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *CoRR*, abs/1611.05244, 2016.

[8] G. Ghiasi, T.-Y. Lin, and Q. V. Le. Dropblock: A regularization method for convolutional networks. *Neural Information Processing Systems*, 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] Z. He, L. Xie, X. Chen, Y. Zhang, Y. Wang, and Q. Tian. Data augmentation revisited: Rethinking the distribution gap between clean and augmented data, 2019.

[11] S. Hou and Z. Wang. Weighted channel dropout for regularization of deep convolutional neural network. *Association for the Advancement of Artificial Intelligence*, 2019.

[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[13] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[14] S. Liu and Z. Liu. Multi-channel cnn-based object detection for enhanced situation awareness. *CoRR*, abs/1712.00075, 2017.

[15] S. N., H. G., K. A., S. I., and S. R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res. 15*, 2014.

[16] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[17] T. Ophoff, K. Van Beeck, and T. Goedemé. Exploring rgb+depth fusion for real-time object detection. *Sensors*, 19:866, 02 2019.

[18] S. Park and N. Kwak. Analysis on the dropout effect in convolutional neural networks. *Asian Conference on Computer Vision*, 2016.

[19] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[20] D. Soekhoe, P. Putten, and A. Plaat. On the impact of data set size in transfer learning using deep neural networks. pages 50–60, 10 2016.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[22] C. Sutton, M. Sindelar, and A. Mccallum. Feature bagging: Preventing weight undertraining in structured discriminative learning. Technical report, 2005.

[23] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada. Multispectral object detection for autonomous vehicles. *Proceedings of the on Thematic Workshops of ACM Multimedia*, pages 35–43, 2017.

[24] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning, 2018.

[25] J. Tompson, A. J. Ross Goroshin, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. *Computer Vision and Pattern Recognition*, pages 1–10, 2015.

[26] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. *ICML*, 2013.

[27] D. Wu, L. Pigou, P. Kindermans, N. D. Le, L. Shao, J. Dambre, and J. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1583–1597, 2016.

[28] H. Wu and X. Gu. Towards dropout training for convolutional neural networks. *Neural Networks*, pages 1–10, 2015.

[29] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar. Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum. Technical report, Nov 2008.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks?, 2014.

[31] A. Zacharopoulos, K. Hatzigiannakis, P. Karamaoynas, V. M. Papadakis, M. Andrianakis, K. Melessanaki, and X. Zabulis. A method for the registration of spectral images of paintings and its evaluation. *Journal of Cultural Heritage*, 29:10 – 18, 2018.

[32] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning, 2019.