Using Error Level Analysis to remove Underspecification

Jérémie Dentan
jeremie.dentan@polytechnique.org
École Polytechnique
France

Figure 1: General overview of the data. There are four types of images, from left to right: Age Young Text Young (AYTY), Age Young Text Old (AYTO), Age Old Text Old (AOTO) and Age Old Text Young (AOTY). The goal is, using only AYTY and AOTO for the training, to be able to predict the age (Young/Old) of image belonging to the four types in the test set.

ABSTRACT

The complexity of recent ML models and their number of weights makes it very difficult to understand the features in the training data that lead to a given model output. This problem results in what is called underspecification: when several very different predictors have similar performances on the training data. As explained in [5], underspecification can have harmful consequences when deploying a model on real data, leading to unexpectedly poor performances.

This challenge addresses this issue by artificially creating a change in distribution between training data and testing data. Thus, the goal is to predict the age (Young/Old) of individuals from photos, but the training data favors learning a predictor based on reading a Young/Old text and not on face analysis. However, this text-based predictor cannot work on the test set, because in the latter text is not correlated with age.

To answer this question, we have tried two approaches.

First, we wanted to implement the DivDis network described by [12]. This network uses several classification heads, which correspond to predictors using different features of the training data. To improve the performance of this model, we have pretrained on purpose a classification head to do precisely what should be avoided, i.e. classify images using only their text. However, this approach was not conclusive and did not lead to good performances.

Second, we have exploited the fact that underspecification here is quite simple and can be bypassed. Thus, we used Error Level Analysis (ELA), a simple method based on JPEG compression, to isolate text areas in the images, in order to remove them, thus solving the underspecification problem.

Although this last method is less in line with the philosophy of the challenge, it allowed us to obtain better accuracy than with the DivDis architecture (73% vs. 64%).

KEYWORDS

Underspecification; Error Level Analysis; Age Prediction.

1 INTRODUCTION

This technical report is part of a data challenge.

- The challenge is available here.
- This report comes with a GitHub repository that contains the source code of all our experiments as well as a link to download the data, available here.
- This report come with a presentation that is available in the doc folder of the repository.

1.1 Description of the problem

The goal of this challenge is to predict the age of people based on pictures of them. It is a binary classification task, since the labels are only Young/Old. However, as shown in figure 1, there is a text inserted on every picture. Depending on the tuple (picture, text), the pictures are classified into four groups, and all of them are not present in the train set:

- Age Young Text Young (AYTY), present in the train/test sets
- Age Young Text Old (AYTO), present in test set only
- Age Old Text Old (AOTO), present in the train/test sets
- Age Old Text Young (AOTY), present in the test set only

The main challenge here is that it is much more simpler for a neural network to learn to read a text inserted in an image than to learn to analyse the face and predict the age. Thus, a simple naive implementation would train a predictor to read the text, which will completely fail when deployed on the test set.

This is the underspecification problem, that is very well described in [5]: several predictors are possible for the train set (here, a predictor based on the text and a predictor based on the face), but some of them behave badly on the test set due to a shift of distribution (here, the apparition of AYTO and AOTY images, that are incompatible with a text-based predictor).

Thus, the goal of this challenge is to find a way to favor the face-based predictor over the text-based predictor, even though the latter is much more simpler to learn.

Rules of the competition. To makes this task harder, this challenge comes with several rules:

- It is forbidden to manually annotate the images of the test set.
- It is forbidden to use data other than those provided by the organizers or the challenge.
- It is forbidden to use pretrained networks other than those trained on ImageNet [6].

Those rules prevent some workaround that would make the challenge easier. Thus, the third rules forbids to use some networks that would be pretrained to predict the age of people's face. Indeed, many papers have explored this tasks, such as [10, 18, 20]; given that those networks were trained on data where the text insertions are absent, there is no underspecification problem, and we can expect those models to have great performances on our test set. Moreover, the second rules prevent the scrapping of a vast amount a images of people whose age are known, which would enable one to train a model on a dataset where the text insertions are absent, nullifying the underspecification problem. Finally, the third rule makes it harder to use out-of-the-box libraries for text detection. Indeed, many text-detection models are pretrained, such as [13, 14], and the widely-used Tesseract OCR library [1] also contains pretrained models.

Descriptions of the datasets. Four datasets were provided by the organizers of the competition. Two were related to the problem of age prediction we described. In addition, the authors provided two datasets that addressed a similar problem, but replacing the prediction of the age by the prediction of the colour of the hair (Dark/Light), resulting in four classes HDTD, HDTL, HLTL, HLTD (equivalent of the AYTY, AYTO, AOTO and AOTY classes for the age). The idea is that prediction of the colour of the hair is much easier for a neural network than prediction of the age, so the participants could test some approaches on the hair datasets before further experiments on the age datasets. Moreover, the labels were available for both train and test set for the hair datasets, making it possible to precisely assess the performance of an approach.

Thus, the datasets were the following:

- `human_hair` : 2000 RGB images of size 178×218 stored in JPEG format:
  - Train set: 1000 labeled images, containing roughly 50% of HDTD and 50% of HLTL
  - Test set: 1000 labeled images, containing roughly 25% of each class
- `human_age` : 89732 RGB images of size 178×218 stored in JPEG format:
  - Train set: 20000 labeled images, containing roughly 50% of AYTY and 50% of AOTO
  - Test set: 69732 unlabeled images; a manual exploration suggesting that it contains roughly 25% of each class

Thus, the dataset are balanced, and our manual explorations suggest that the test set is balanced as well. Given that the evaluation metric is the accuracy of the age prediction, we did not have to do any data augmentation step on those balanced datasets.

1.2 Previous work on underspecification and robustness to ambiguity

Underspecification and ambiguity are two similar concepts, and we will use these two terms equivalently here. These two notions refer to a situation where several predictors can have similar performances on training data, while being very different. This can have negative consequences both in terms of explainability and robustness to a change in distribution.

The problem of underspecification have been characterized quite early, in 2016, by [2]. Then, in 2020, several paper discussed this issue on several datasets [5, 19]. Moreover, some papers have shown that neural networks prioritize simple patterns during the training, which is the origin of the underspecification problem [4, 8].

Moreover, many papers have focused on the robustness of models in a broad sense, such as [3, 7, 15, 17, 21, 25]. However, these methods address the robustness problem from the perspective of a distribution shift. Thus, although the motivation is the same, these methods, which only learn a single predictor, are inefficient when the objective function is really ambiguous as it is the case for this challenge.

A solution to the ambiguity problem, called the DivDis architecture, has been proposed very recently, in 2022, by [12]. The idea of this paper is to train several predictors on the same problem and on the same training data set, favoring predictors that contradict each other on a test data set. Thus, this paper makes the assumption that the objective function of the training data is really ambiguous with respect to the distribution on the test set. Thus, by forcing the different predictors to contradict each other on the test data, it is hoped that these predictors will learn different aspects of the training data.

1.3 Discussion on the DivDis architecture and justification of our approach

The approach of [12], with the DivDis architecture, can be criticized. Indeed, it makes the strong assumption that we have two datasets for which we know that the distributions are different, which justifies the fact of gratifying during training the classifiers that contradict each other on the test data.

For example, for the `human_age` dataset, we know that in the train set, the objective function of the age is ambiguous between classes AYTO and AYTY on the one hand, and between classes AOTO and AOTY on the other hand. Thus, with the architecture of [12], we hope to train two predictors, one that would be text-based, and one that would be face-based; and those predictors would nicely contradict each other on the test set.

However, in many real-world cases, knowing for sure that the distribution is different between two datasets, so that we can assume that the predictors will contradict each other, is not far from knowing where the ambiguity comes from and what it consists of. Moreover, given that this architecture trains several predictors, there is the need of an expert that have domain knowledge to choose between the predictors and decide which one is ambiguity-free and can be deployed.

However, as said above, underspecification comes from the fact that neural network tend over-rely on simple features, that are easy to detect, and thus likely to be detectable by humans. Thus, the same expert that have domain knowledge to decide which head to finally use is likely to also have the domain knowledge to understand where the ambiguity comes from and how to characterize it.

This is why we have explored two ways for solving this challenge:

- One using the architecture of [12], adapted to our problem by pretraining a predictor on the text (by using both the `human_hair` and `human_age` datasets). Unfortunately, we did not complete this method because the predictor we wanted to pretrain for the text did not perform as expected.
- The other that presupposes the existence of an expert with domain knowledge, and that explores simple numerical methods that effectively remove ambiguity. In this case, we used the JPEG compression and Error Level Analysis (ELA) to efficiently spot the text insertions and remove them.

2 OUR APPROACH

As explained in section 1.3, we worked on two different approaches to solve this challenge: the first one using the architecture of [12], which did not succeed, and the second one using ELA to spot the text insertions and remove them.

2.1 First approach: training a predictor to read the text

In this section, we describe how we tried to use both the `human_hair` and `human_age` datasets to train a predictor to read the text on an image, thus adapting the DivDis architecture [12] to our challenge.

Indeed, the baseline provided by the organizers of the challenge already implemented the DivDis architecture, so if we wanted to beat its performances, we needed to adapt it specifically to our classification task.

As explained in section 1.3, the architecture of DivDis trains several predictors while ensuring they disagree on the test set. In our case, this mean training two predictors, hopping one would learn to read the text and one would learn to understand the age directly from the faces (cf. figure 2).

Thus, to improve the performance of this architecture, we wanted to force one of the predictor to learn to read the text. This cannot simply be done using the `human_age` dataset: indeed, in that dataset, the text "Young"/"Old" is known to be correlated to another feature, which is the age of the face on the picture. However, if we mix the datasets `human_hair` and `human_age` , the text is no longer fully correlated to any other features. Indeed, the `human_hair` dataset contains pictures of both old and young people where the text is neither "Old" nor "Young", and similarly the `human_age` dataset contains pictures of both people with dark and light hairs where the text is neither "Dark" nor "Light". Thus, if a predictor is trained to predict one of the four classes "Young", "Old", "Dark" "Light" on those two datasets, it is expected to learn to read the text on the image, and not to analyse the face.

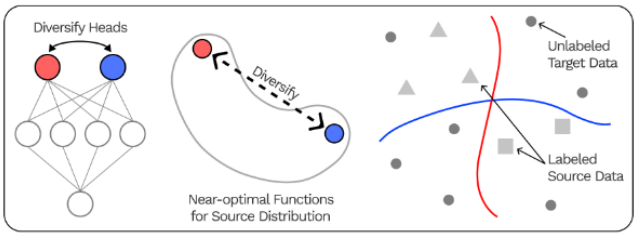

Figure 2: Figure taken from [12]: general principle of the DivDis architecture. Two different predictors are trained, with two different separating functions (the blue and red lines), which lead to point they disagree about in the test set.

More precisely, we did the following:

- We used the pretrained `ResNet` [9] architecture (which is allowed since it is trained on ImageNet [6]) to have a network that is already trained to detect features on natural images.
- We froze the first layers of `ResNet`, and trained only the last ones to predict one of the four classes: "Young", "Old", "Dark" "Light"
- We shuffled the `human_age` train set and the `human_hair` train set, which are all labeled. For those datasets, we know that the text insertion matches the label Dark/Light or Young/Old, so we have access to the true label of the text. This is not the case for the test set of `human_hair` : we only have access to the Dark/Light label, but for each image we do not know if the text insertion matches the label.
- We separated 20% of our data to be a validation set, and we implemented early stopping on our training pipeline to avoid overfitting
- To improve the size of our datasets, we used data replications. The transformations we randomly added when replicating images were:
  - Adding a Gaussian noise
  - "Rolling" the RGB colours (e.g. RGB → BRG, etc). Indeed, reading the text is independent of the colour, but on the other hand the features that might be correlated to the text, i.e. the hair colour or the age, are colour-dependent. Moreover, "Dark" and "Light" were always written in the same colour, so we needed to decorrelate this.
  - Reverting the lightness (i.e. each pixels gets its opposite value) to make sure the network does not learn any colour from the image
- We used a replication factor of 20 for the `human_hair` dataset and of 2 for the `human_age` dataset, resulting in a balanced dataset between our four classes.

With this pipeline, we obtained an accuracy of 95% on our validation set. However, some manual checks on the test set of `human_age` showed that our network did not correctly learnt to read the text. This is probably linked to some correlations between the text and the faces that we did not correctly deleted in our training data, and that caused the poor performances on the `human_age` test set. Indeed, even though we did our best to remove correlation between the text and the face, there is a huge distribution shift between the union of train sets of `human_age` and `human_hair` on the one hand, and the test set of `human_age` on the other hand, and this huge distribution shift probably caused this unexpected poor performances.

Due to those poor performances of the predictor we wanted to teach to read the text on the images, we decided to abandon this approach, and switched to the one that is presented in the following section.

2.2 Second approach: using ELA to remove the text insertions

In this section, we describe how we successfully used Error Level Analysis (ELA) to detect the text insertions on the images and remove them, enabling us to train a classifier on unambiguous data.

2.2.1 Detecting the text with ELA. We wanted to use a simple numerical method to detect the text insertions and remove them. Indeed, many Optical Character Recognition (OCR) methods exist to detect and even read text on images [1, 13, 14], however they present the following problems for this competition:

- The majority of recent OCR method rely on pretrained models, so we are not allowed to use them for the competition
- For some images, there is text in the background, that we do not want to remove, and that would be removed by out-of-the-box OCR. We only want to remove the text insertions "Young" and "Old".

This is why we decided to develop a customized method to detect the text on our data. Given that the text is artificially inserted in the pictures, we explored several forgery detection methods. This task is made difficult by several factors:

- The text is written in different colour and fonts, so there is not a unique pattern "young" and a unique pattern "old"
- The contrast between the text insertions and the background is not abnormally high compared to the contrast in other regions of the pictures
- There is noise added in the text, so the letters are not formed of a continuous area having the same colour
- Many font do not present any straight edge, so we cannot use edge detectors
- The number of pixel of the image is pretty low, which puts in difficulty a certain number of forgery detection techniques

However, after having tested several forensic data analysis techniques [16], we found that the best one for our problem we Error Level Analysis (ELA), that have already been used for forgery detection, for example by [24]. The principle of ELA is to measure the mean reconstruction error among the 3 colour channel after a JPEG compression of a given quality factor [23]. The use of this technique for forgery detection assumes that the forged area will have a different level of error than the rest of the image, making it detectable.

Using this technique, our text detection pipeline was the following (cf. figure 3):

- Do a JPEG compression of quality 0.95 (this hyperparameter was optimized with several tests);
- Compute the distance of every pixel between the compressed image and the original, and then take the mean among the 3 colour channels;
- Remove noise by applying a moving average with a 20×20 window and then apply a threshold by setting to 0 every pixel whose value is less than 0.7
- Take the barycenter of the remaining pixels
- Place a rectangle of fixed size whose center is the barycenter. We use a fixed size to avoid any bias linked to the fact that word "Young" is longer than word "Old".

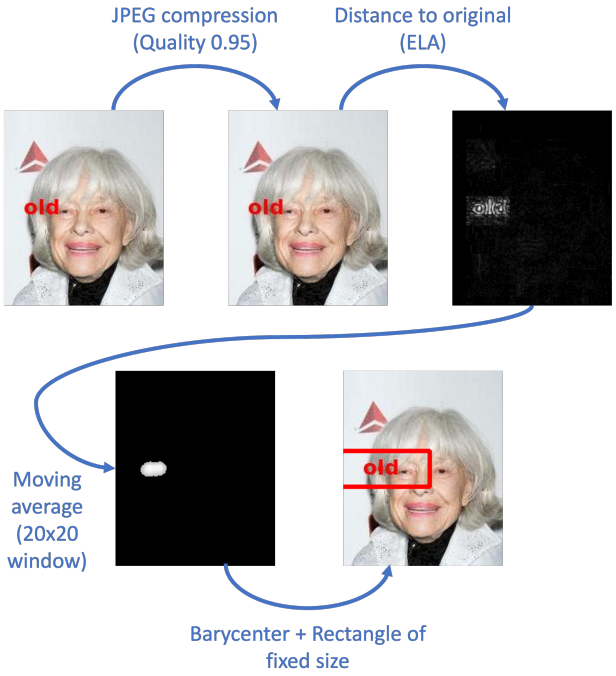

Figure 3: Our pipeline to detect the text and remove it, using JPEG compression and ELA

Performances of our text extractor. This approach worked quite well, and the text extractor we described resulted in sufficient performance to remove ambiguity from our data. Thus, after a manual check on 200 images, we found that the text extractor **worked perfectly in 91% of the cases:**

- On 182 images out of 200, the text was entirely inside the detection rectangle, even when other textual elements were present on the image (cf figure 3). Indeed, the parameters of the ELA method that we applied were optimized to detect forgery images, whereas a text naturally present on an image is not necessarily a forgery.

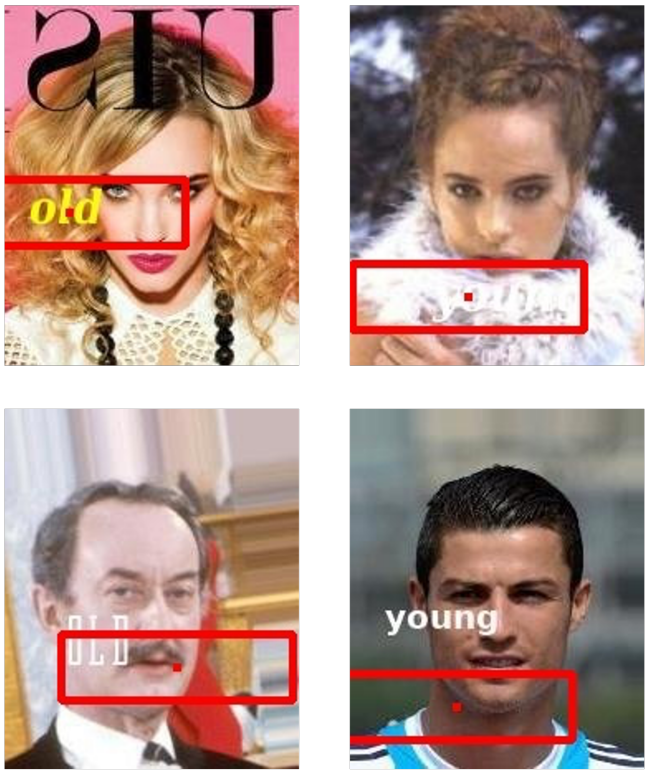

Figure 4: Some examples of text extraction. Some manual checks on 200 images shows that the extractor fully misses it target in only 5% of the images, and misses it partially in 4% of them. Up-left: the text extractor detects the true text insertion rather than the text already on the picture; Up-right: the text extractor detects the text even with very low contrast (the text "young" is written in white); Down-left: the text extractor partially misses its target (at least one part of a letter is outside the rectangle); Down-right: the text extractor completely misses its target (most of the text is outside the rectangle)

- On 8 images out of 200, the text was partially outside the detection rectangle
- On 10 images out of 200, the text was mostly outside the detection rectangle.

Thus, once the detection rectangle is removed, a predictor has little chance of learning to read text insertions on images, as these are only useful in 9% of the training data.

This is why, although we still have a margin of progress on this text extractor, this performance of 91% seemed to us sufficient so that once the detection rectangles are removed, there is no more strong correlation between the age of the faces and the text appearing on the image. Thus, we considered that the ambiguity of our training data was removed.

Computation time. For our experiments, we used a Intel Xeon W-1290P 3.70GHz with 10 physical cores, and we distributed the processing of the 89732 images among the 20 virtual cores of the processor. With the technique we described, it took 455s to compute the barycenters of the detection rectangle for all images, i.e. about 10 images per second and per process.

2.2.2 Training a network to predict the age after the removal of the text. After having removed the detection rectangle where the text is located in 91% of the images (cf. section 2.2.1), thus removing the ambiguity of the data, our task was much simpler. Indeed, we simply had to implement a classification pipeline using a pretrained network on ImageNet [6]. However, the task of predicting the age of people based on the analysis of their face remains challenging, so we had to optimize as much as possible our training pipeline. To do so, we were inspired by the choices made in the pipelines of [20] and [10]. We did the following:

- We tested two pretrained networks, VGG19 [22] and ResNet50 [9], which are both well-recognized and often used within the computer vision community.
- We set to 0 the value of every pixel within the detection rectangle of the text
- We used the well recognized Adam optimizer [11] with a binary cross-entropy loss and a batch size of 32.
- We used gradual unfreezing of the layers: first, 2 epochs with learning rate 10^{-4} training only the classification head, then unfreezing the last convolutional layers (layer 4 for ResNet50, containing 9 convolutions, or convolutions 28 to 34 for VGG) and doing 3 epochs with learning rate 5.10^{-5}, then unfreezing one additional layer (layer 3 for ResNet50, containing 18 convolutions, or convolutions 19 to 27 for VGG) and doing 3 epochs with learning rate $2.5.10^{-5}$, and finally unfreezing all the network and doing 5 epochs with learning rate 2.10^{-5}
- We used a validation set of 20% of the data and early stopping with patience 3 epochs plus memorization of the best validation performance to avoid overfitting
- We used data replication with a replication factor of 3 to improve the size of our train set. When replicating the images, we randomly added Gaussian noise of variance 10% of the range of the pixel values.

For more details about our pipeline, please refer to the source code of our experiments, which is available here.

Performances and training time. This pipeline enable us to reach an accuracy of 73%, which is quite satisfying given that the baseline with the DivDis architecture [12] obtained an accuracy of 64%. Our GPU computations were done on a NVIDIA GeForce RTX 3090 24Go. The training and prediction with the VGG architecture took about 760s per epoch, i.e. 9880s (around 2h45) for the 13 epochs if the early stopping is not triggered. The training with the ResNet architecture took a similar amount of time.

3 CONCLUSION

Developing models such as DivDis [12] that can fight underspecification in various setting seems promising, however this approach is hampered by the need to have an expert with sufficient knowledge to evaluate the model on the test data and choose the most suitable predictor.

To solve this challenge, our final approach consisted in removing the areas of the images concerned by underspecification, which

corresponds to a situation where the expert manages to characterize the ambiguity of the data so that it can be removed.

To do this, we used a fairly simple numerical method based on JPEG conversion, which fully removed the ambiguity in 91% of the cases. Moreover, our approach shows that forensic analysis can be useful to efficiently isolate specific areas in images, with a much lower computation cost than using a neural network.

This approach allowed us, after using a fine-tuned ResNet50 [9] network on our data, to achieve an accuracy of 73%, thus beating the baseline of 64% that was obtained with the DivDis architecture [12].

REFERENCES

[1] 2023. Tesseract OCR. https://github.com/tesseract-ocr/tesseract original-date: 2014-08-12T18:04:59Z.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. https://doi.org/10.48550/arXiv.1606.06565 arXiv:1606.06565 [cs].

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2020. Invariant Risk Minimization. https://doi.org/10.48550/arXiv.1907.02893 arXiv:1907.02893 [cs, stat].

[4] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A Closer Look at Memorization in Deep Networks. https://doi.org/10.48550/arXiv.1706.05394 arXiv:1706.05394 [cs, stat].

[5] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2020. Underspecification Presents Challenges for Credibility in Modern Machine Learning. https://doi.org/10.48550/arXiv.2011.03395 arXiv:2011.03395 [cs, stat].

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848 ISSN: 1063-6919.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. https://doi.org/10.48550/arXiv.1505.07818 arXiv:1505.07818 [cs, stat].

[8] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. 2019. Implicit Bias of Gradient Descent on Linear Convolutional Networks. https://doi.org/10.48550/arXiv.1806.00468 arXiv:1806.00468 [cs, stat].

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. http://arxiv.org/abs/1512.03385 arXiv:1512.03385 [cs].

[10] Shakediel Hiba and Yosi Keller. 2021. Hierarchical Attention-based Age Estimation and Bias Estimation. http://arxiv.org/abs/2103.09882 arXiv:2103.09882 [cs] version: 1.

[11] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980 arXiv:1412.6980 [cs].

[12] Yoonho Lee, Huaxiu Yao, and Chelsea Finn. 2022. Diversify and Disambiguate: Learning From Underspecified Data. https://doi.org/10.48550/arXiv.2202.03418 arXiv:2202.03418 [cs, stat].

[13] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2022. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. https://doi.org/10.48550/arXiv.2109.10282 arXiv:2109.10282 [cs].

[14] Junyang Lin, Xuancheng Ren, Yichang Zhang, Gao Liu, Peng Wang, An Yang, and Chang Zhou. 2022. Transferring General Multimodal Pretrained Models to Text Recognition. https://doi.org/10.48550/arXiv.2212.09297 arXiv:2212.09297 [cs].

[15] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just Train Twice: Improving Group Robustness without Training Group Information. https://doi.org/10.48550/arXiv.2107.09044 arXiv:2107.09044 [cs, stat].

[16] Jörg Meyer. 2012. *Forensische Datenanalyse : dolose Handlungen im Unternehmen erkennen und aufdecken.* Erich Schmidt. Google-Books-ID: V96iMAEACAAJ.

[17] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from Failure: Training Debiased Classifier from Biased Classifier. https://doi.org/10.48550/arXiv.2007.02561 arXiv:2007.02561 [cs, stat].

[18] Cao Hong Nga, Khai-Thinh Nguyen, Nghi C. Tran, and Jia-Ching Wang. 2020. Transfer Learning for Gender and Age Prediction. In *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*. 1–2. https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258347 ISSN: 2575-8284.

[19] Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. 2019. Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging. https://doi.org/10.48550/arXiv.1909.12475 arXiv:1909.12475 [cs, stat].

[20] Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D. Barkana. 2017. Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model. http://arxiv.org/abs/1709.01664 arXiv:1709.01664 [cs].

[21] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. https://doi.org/10.48550/arXiv.1911.08731 arXiv:1911.08731 [cs, stat].

[22] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556 arXiv:1409.1556 [cs].

[23] A. Skodras, C. Christopoulos, and T. Ebrahimi. 2001. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine* 18, 5 (Sept. 2001), 36–58. https://doi.org/10.1109/79.952804 Conference Name: IEEE Signal Processing Magazine.

[24] Ida Sudiatmika, Fathur Rahman, Trisno Trisno, and Suyoto Suyoto. 2018. Image forgery detection using error level analysis and deep learning. *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 17 (Aug. 2018), 653. https://doi.org/10.12928/telkomnika.v17i2.8976

[25] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. https://doi.org/10.48550/arXiv.1412.3474 arXiv:1412.3474 [cs].

Received 21 March 2023