REFERENSI

[1] Ahmad, H. M., & Rahimi, A. (2022). Deep learning methods for object detection in smart manufacturing: A survey. Journal of Manufacturing Systems, 64, 181-196.

[2] Mittal, S., Khan, M. A., Romero, D., & Wuest, T. (2019). Smart manufacturing: Characteristics, technologies and enabling factors. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 233(5), 1342-1361.

[3] Zheng, P., Wang, H., Sang, Z., Zhong, R. Y., Liu, Y., Liu, C., Mubarok, K., Yu, S., & Xu, X. (2018). Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. Frontiers of Mechanical Engineering, 13(2), 137-150.

[4] Zhang, C., Shi, W., Li, X., Zhang, H., & Liu, H. (2018). Improved bare PCB defect detection approach based on deep feature learning. The Journal of Engineering, 2018(16), 1415-1420.

[5] Tang, S., He, F., Huang, X., & Yang, J. (2019). Online PCB defect detector on a new PCB defect dataset. arXiv preprint arXiv:1902.06197.

[6] Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. Pattern recognition, 34(12), 2259-2281.

[7] Hoy, D. E. (1997). On the use of color imaging in experimental applications. Experimental Techniques, 21(4), 17-19.

[8] Pietikainen, M., Nieminen, S., Marszalec, E., & Ojala, T. (1996, August). Accurate color discrimination with classification based on feature distributions. In Proceedings of 13th International Conference on Pattern Recognition (Vol. 3, pp. 833-838). IEEE.

[9] Littmann, E., & Ritter, H. (1997). Adaptive color segmentation-a comparison of neural and statistical methods. IEEE Transactions on neural networks, 8(1), 175-185.

[10] Garcia-Lamont, F., Cervantes, J., López, A., & Rodriguez, L. (2018). Segmentation of images by color features: A survey. Neurocomputing, 292, 1-27.

[11] Anderson, S., Pringle, M., Eadie, M., Austin, T., Wilson, A., & Polfreman, M. (2006). Digital images archiving study. Arts and Humanities Data Service,¡ wwwjisc. ac. uk/uploaded_documents/FinaldraftImagesArchivingStudy. pdf.

[12] Oppenheim, A. V., Willsky, A. S., Nawab, S. H., & Ding, J. J. (1997). Signals and systems (Vol. 2, pp. 74-102). Upper Saddle River, NJ: Prentice hall.

[13] Lipschitz, R. (1876). Sur la possibilité d'intégrer complètement un système donné d'équations différentielles. Bulletin des sciences mathématiques et astronomiques, 10, 149-159.

[14] Picard, É. (1890). Memoire sur la theorie des equations aux derivees partielles et la methode des approximations successives. Journal de Mathématiques pures et appliquées, 6, 145-210.

[15] Lindelöf, E. (1894). Sur l'application des méthodes d'approximations successives à l'étude des intégrales réelles des équations différentielles ordinaires. Journal de mathématiques pures et appliquées, 10, 117-128.

[16] Wang, H., & Raj, B. (2017). On the origin of deep learning. arXiv preprint arXiv:1702.07800.

[17] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

[18] Bridle, J. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. Advances in neural information processing systems, 2.

[19] Nair, V., & Hinton, G. E. (2010, January). Rectified linear units improve restricted boltzmann machines. In Icml.

[20] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science.

[21] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[22] Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. Comp. Rend. Sci. Paris, 25(1847), 536-538.

[23] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

[24] Bera, S., & Shrivastava, V. K. (2020). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. International Journal of Remote Sensing, 41(7), 2664-2683.

[25] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[26] LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. In Shape, contour and grouping in computer vision (pp. 319-345). Springer, Berlin, Heidelberg.

[27] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[28] Laserlicht, M. H. (2019, July 3). Raspberry Pi 4 Model B from the side. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Raspberry_Pi_4_Model_B_-_Side.jpg

[29] Raspberry Pi Foundation. (n.d.). [Completely upgraded, re-engineered Faster, more powerful]. Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-4-model-b

[30] Debian Project. (n.d.-a). Debian – Reasons to use Debian. Retrieved May 30, 2023, from https://www.debian.org/intro/why_debian

[31] Raspberry Pi Foundation. (2020, May 28). The logo of Raspberry Pi OS. https://www.raspberrypi.org/documentation/

[32] Python Software Foundation. (n.d.-c). PEP 8 – Style Guide for Python Code — peps.python.org. Python Enhancement Proposals. Retrieved June 18, 2023, from https://peps.python.org/pep-0008/

[33] Python Software Foundation. (n.d.-b). Glossary. Python Documentation. Retrieved June 18, 2023, from https://docs.python.org/3/glossary.html#term-parameter

[34] Python Software Foundation. (n.d.). 9. Classes — Dokumentasi Python 3.8.16. Python Documentation. Retrieved June 20, 2023, from https://docs.python.org/id/3.8/tutorial/classes.html#class-and-instance-variables

[35] Pyhton Software Foundation. (n.d.). Ikhtisar — Dokumentasi Python 3.8.16. Python Documentation. Retrieved June 20, 2023, from https://docs.python.org/id/3.8/glossary.html#term-decorator

[36] Huyen, C. (2022). Machine Learning Systems Design. Stanford CS 329S. Retrieved June 21, 2023, from https://stanford-cs329s.github.io/syllabus.html

[37] Ajay, M., Sharma, A., Goyal, N., & Deshpande, A. (2022, January 19). Reliable Machine Learning - Microsoft Research. Microsoft Research. Retrieved June 21, 2023, from https://www.microsoft.com/en-us/research/group/reliable-machine-learning/

[38] Saria, S., & Subbaswamy, A. (2019). Tutorial: safe and reliable machine learning. arXiv preprint arXiv:1904.07204.

[39] DeepLearningAI, & Ng, A. (2021, March 24). A Chat with Andrew on MLOps: From Model-centric to Data-centric AI [Video]. YouTube. https://www.youtube.com/watch?v=06-AZXmwHjo

[40] Subbaswamy, A., Schulam, P., & Saria, S. (2019, April). Preventing failures due to dataset shift: Learning predictive models that transport. In The 22nd International Conference on Artificial Intelligence and Statistics (pp. 3118-3127). PMLR.

[41] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[42] Sinha, A., Namkoong, H., Volpi, R., & Duchi, J. (2017). Certifying some distributional robustness with principled adversarial training. arXiv preprint arXiv:1710.10571.

[43] Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., & Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86-92.

[44] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019, January). Model cards for model reporting. In Proceedings of the conference on fairness, accountability, and transparency (pp. 220-229).

[45] Huyen, C. (2019, November 25). Machine Learning Interviews. GitHub. Retrieved June 22, 2023, from https://github.com/chiphuyen/machine-learning-systems-design/blob/master/build/build1/consolidated.pdf

[46] Ding, J., Tarokh, V., & Yang, Y. (2018). Model selection techniques: An overview. IEEE Signal Processing Magazine, 35(6), 16-34.

[47] Ameisen, E. (2020). Building Machine Learning Powered Applications: Going from Idea to Product. " O'Reilly Media, Inc.".

[48] Geron, A. (2017). Hands-On Machine Learning with Scikit-Learn & Tensorflow O'Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472, 564.

[49] Wolf, M. J., Miller, K., & Grodzinsky, F. S. (2017). Why we should have seen that coming: comments on Microsoft's tay" experiment," and wider implications. Acm Sigcas Computers and Society, 47(3), 54-64.

[50] Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., ... & Heidmann, L. (2020). Introducing MLOps. O'Reilly Media.

[51] Google Developers. (2023, March 3). Understand the problem. Google for Developers. Retrieved June 30, 2023, from https://developers.google.com/machine-learning/problem-framing/problem

[52] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in neural information processing systems, 28.

[53] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Young, M. (2014). Machine learning: The high interest credit card of technical debt.

[54] Cunningham, W. (1992). The WyCash portfolio management system. ACM Sigplan Oops Messenger, 4(2), 29-30.

[55] Rogati, M. (2017, June 12). The AI Hierarchy of Needs. HackerNoon. Retrieved July 7, 2023, from https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

[56] Raj, E. (2021). Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale. Packt Publishing Ltd.

[57] Moroney, L. (2019). Horses or Humans Dataset (3.0.0) [Dataset]. http://laurencemoroney.com/horses-or-humans-dataset

[58] Degli Esposti, M., Lagioia, F., & Sartor, G. (2020). The use of copyrighted works by AI systems: Art works in the data mill. European Journal of Risk Regulation, 11(1), 51-69.

[59] Sobel, B. L. (2017). Artificial intelligence's fair use crisis. Colum. JL & Arts, 41, 45.

[60] He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., ... & Candela, J. Q. (2014, August). Practical lessons from predicting clicks on ads at facebook. In Proceedings of the eighth international workshop on data mining for online advertising (pp. 1-9).

[61] Melis, G., Dyer, C., & Blunsom, P. (2017). On the state of the art of evaluation in neural language models. arXiv preprint arXiv:1707.05589.

[62] Karayev, S., Tobin, J., & Abbeel, P. (2021). Lecture 7: Troubleshooting Deep Neural Networks. The Full Stack. Retrieved July 15, 2023, from https://fullstackdeeplearning.com/spring2021/lecture-7/

[63] Google Developers. (2022, January 19). Menggunakan format Model Tersimpan. TensorFlow. Retrieved July 21, 2023, from https://www.tensorflow.org/guide/saved_model?hl=id

[64] Walber. (2014, November 22). Precision and recall. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Precisionrecall.svg

[65] Oh, S., Kim, M., Kim, D., Jeong, M., & Lee, M. (2017, August). Investigation on performance and energy efficiency of CNN-based object detection on embedded device. In 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT) (pp. 1-4). IEEE.

[66] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).

[67] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[68] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE-/CVF Conference on Computer Vision and Pattern Recognition (pp. 2820-2828).

[69] Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. arXiv preprint arXiv:1912.06059.

[70] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[71] Gang, S., Fabrice, N., Chung, D., & Lee, J. (2021). Character recognition of components mounted on printed circuit board using deep learning. Sensors, 21(9), 2921.

[72] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[73] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[74] Sifre, L., & Mallat, S. (2014). Rigid-motion scattering for texture classification. arXiv preprint arXiv:1403.1687.

[75] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.

[76] Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization?. Advances in neural information processing systems, 31.

[77] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. Neural networks, 13(4-5), 411-430.

[78] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Swish: a self-gated activation function. arXiv preprint arXiv:1710.05941, 7(1), 5.

[79] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).

[80] Chiu, Y. C., Tsai, C. Y., Ruan, M. D., Shen, G. Y., & Lee, T. T. (2020, August). Mobilenet-SSDv2: An improved object detection model for embedded systems. In 2020 International conference on system science and engineering (ICSSE) (pp. 1-5). IEEE.

[81] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[82] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[83] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[84] Ghiasi, G., Lin, T. Y., & Le, Q. V. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7036-7045).

[85] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[86] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat, 37, 547-579.

[87] Qin, Z., Li, Z., Zhang, Z., Bao, Y., Yu, G., Peng, Y., & Sun, J. (2019). ThunderNet: Towards real-time generic object detection on mobile devices. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6718-6727).

[88] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European conference on computer vision (ECCV) (pp. 116-131).

[89] Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2017). Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264.

[90] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29.

[91] Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. Proceedings of the IEEE.

[92] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence, 38(1), 142-158.

[93] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., ... & Walsh, J. (2020). Deep learning vs. traditional computer vision. In Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1 (pp. 128-144). Springer International Publishing.

[94] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[95] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338.

[96] Everingham, M., Eslami, S. M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98-136..

[97] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[98] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.

[99] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., ... & Lin, D. (2019). MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155.

[100] PyTorch Mobile. (n.d.). PyTorch. Retrieved August 10, 2022, from https://pytorch.org/mobile/home/

[101] Deploy machine learning models on mobile and edge devices. (n.d.). Tensorflow. Retrieved August 10, 2022, from https://www.tensorflow.org/lite

[102] Quickstart for Linux-based devices with Python. (n.d.). Tensorflow. Retrieved August 10, 2022, from https://www.tensorflow.org/lite/guide/python

[103] Object detection. (n.d.). Tensorflow. Retrieved August 10, 2022, from https://www.tensorflow.org/lite/examples/object_detection/overview

[104] Padilla, R., Netto, S. L., & Da Silva, E. A. (2020, July). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE.

[105] Gauen, K., Rangan, R., Mohan, A., Lu, Y. H., Liu, W., & Berg, A. C. (2017, January). Low-power image recognition challenge. In 2017 22nd Asia and South pacific design automation conference (ASP-DAC) (pp. 99-104). IEEE.

[106] Alyamkin, S., Ardi, M., Brighton, A., Berg, A. C., Chen, Y., Cheng, H. P., ... & Zhuo, S. (2018). 2018 low-power image recognition challenge. arXiv preprint arXiv:1810.01732.

[107] Lu, Y. H. (2019). Low-power image recognition. Nature Machine Intelligence, 1(4), 199-199.

[108] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[109] Hu, X. & Low Power Computer Vision. (n.d.). Public Leaderbord. LPCV. Retrieved August 13, 2022, from https://lpcv.ai/scoreboard/Video21

[110] Hu, X., Jiao, Z., Kocher, A., Wu, Z., Liu, J., Davis, J. C., ... & Lu, Y. H. (2023). Evolution of Winning Solutions in the 2021 Low-Power Computer Vision Challenge. Computer, 56(8), 28-37.

[111] Hu, X., Kocher, A., & Jiao, Z. (2021). 21LPCVC-UAV_VIdeo_Track-Sample-Solution (1.0.0) [Software]. Github. https://github.com/lpcvai/21LPCVC-UAV_VIdeo_Track-Sample-Solution

[112] Axiom Test. (n.d.). Yokogawa WT310 Digital Power Meter, DC - 100 kHz, 20A, 1 Ch. Axiom. https://www.axiomtest.com/Electrical-Test-and-Power-Analyzers/Power-Analyzers/Yokogawa/WT310/Digital-Power-Meter,-DC-_-100-kHz,-20A,-1-Ch./

[113] Read, P., & Meyer, M. P. (2000). Restoration of motion picture film. Elsevier.

[114] Google Developers. (2022b, May 26). TensorFlow Lite Model Maker. TensorFlow. Retrieved August 14, 2023, from https://www.tensorflow.org/lite/models/modify/model_maker

[115] Google Developers. (2022c). tflite-model-maker (0.4.2) [Software]. Github. https://github.com/tensorflow/examples/blob/tflmm/v0.4.2/tensorflow_examples/lite/model_maker/public/__init__.py

[116] Google Developers. (2021). EfficientDet [Software]. Github. https://github.com/tensorflow/examples/tree/master/tensorflow_examples/lite/model_maker/third_party/efficientdet

[117] Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics, 462-466.

[118] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[119] Huber, P. J. (1992). Robust estimation of a location parameter. In Breakthroughs in statistics: Methodology and distribution (pp. 492-518). New York, NY: Springer New York.

[120] Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

[121] LeViet, K. & TensorFlow. (2021, December 8). Introduction to object detection on Raspberry Pi [Video]. YouTube. Retrieved August 30, 2023, from https://www.youtube.com/watch?v=mNjXEybFn98

[122] Jocher, G. (202 C.E.). YOLOv5 by Ultralytics (7.0) [Software]. Github. https://github.com/ultralytics/yolov5

[123] Perusahaan Listrik Negara. (2021). Rencana Usaha Penyediaan Tenaga Listrik PT PLN 2021-2023. In PLN (188.K/HK.02/MEM.L/2021). Kementerian Energi dan Sumber Daya Mineral. Retrieved September 11, 2023, from https://web.pln.co.id/statics/uploads/2021/10/ruptl-2021-2030.pdf

[124] McConnell, J. R., & Edwards, R. (2008). Coal burning leaves toxic heavy metal legacy in the Arctic. Proceedings of the national academy of sciences, 105(34), 12140-12144.

[125] Ma, Q., Cai, S., Wang, S., Zhao, B., Martin, R. V., Brauer, M., ... & Burnett, R. T. (2017). Impacts of coal burning on ambient PM 2.5 pollution in China. Atmospheric Chemistry and Physics, 17(7), 4477-4491.

[126] Ando, M., Tadano, M., Asanuma, S., Tamura, K., Matsushima, S., Watanabe, T., ... & Cao, S. (1998). Health effects of indoor fluoride pollution from coal burning in China. Environmental Health Perspectives, 106(5), 239-244.