Deep Video Stream Information Analysis and Retrieval: Challenges and Opportunities

N. Passalis[1], M. Tzelepi[1], P. Charitidis[2], S. Doropoulos[2], S. Vologiannidis[3] and A. Tefas[1]
[1]Dept. of Informatics, Aristotle University of Thessaloniki, Greece
[2]DataScouting, Greece
[3]Dept. of Computer, Informatics and Tel. Engineering, International Hellenic University, Greece
E-mails:{passalis, mtzelepi}@csd.auth.gr,
{pcharitidis, doro}@datascouting.com, svol@ihu.gr, tefas@csd.auth.gr

Abstract

Deep Learning (DL) provided powerful tools for various visual information analysis and retrieval tasks, outperforming previously used methods. However, despite the potential of such approaches for various tasks, applying them in video stream applications, such as media monitoring or surveillance, where a large number of streams should be processed in parallel, is not trivial and comes with several challenges. This paper aims to provide a brief overview of the current state-of-the-art in DL tools that can be used for deep video stream information analysis and retrieval. Apart from a review of the current literature, we also include experimental results discussing deployment challenges, ranging from speed to energy consumption, demonstrating the capabilities of readily available commodity hardware in processing video streams for selected DL models.

1. Introduction

Recent advances in Deep Learning (DL) led to a wide portfolio of tools for various powerful visual information analysis tasks, outperforming traditionally used methods [18]. These range from human detection [17] and recognition [52], as well as object detection approaches [23] to sophisticated emotion [26] and activity recognition approaches [53]. At the same time, powerful DL-based representation learning methods significantly increased the accuracy of information retrieval [27], while also allowed for accurately performing challenging tasks, such as re-identification [60]. However, despite the great potential of such approaches, applying them in real-world scenarios often remains especially challenging. DL methods typically require a different training and deployment pipeline compared to traditional computer vision approaches, while at the same time they often critically rely on the availability of hardware DL accelerators. Indeed, modern DL methods are computationally intensive both to train and deploy requiring a significant amount of equipment and energy for deployment, which can slow down their adoption [40].

These limitations are even more critical in video stream applications [38], such as media analysis and surveillance. In such applications, a large number of parallel streams should be processed in almost real-time in order to extract, consolidate and then provide the necessary information to the end-users. The amount of data that can be gathered in such applications is enormous. A media monitoring application might need to cover tens or hundreds of channels, while CCTV applications in large premises, such as malls, might also scale to hundreds of cameras that need to be simultaneously monitored.

This paper aims to provide a brief overview of the current state-of-the-art in DL tools that can be used for deep video stream information analysis and retrieval, focusing especially on challenges that often arise when processing a large number of video streams. To this end, apart from a review of the current literature, we also include experimental results where we demonstrate the capabilities of commodity hardware in processing video streams for selected DL models. The provided analysis goes beyond traditional metrics, such as memory and speed (e.g., frames-per-second, FPS), by also including energy consumption metrics in order to also take into account this important aspect which can be a limiting factor in many large-scale deployments. The code used for experimental evaluation is available at `https://github.com/cidl-auth/stream-benchmark`.

The rest of the paper is structured as follows. Section 2 provides an overview and critical discussion of currently available DL tools for visual information analysis and retrieval that concern video streams. Then, in Section 3 we provide an experimental analysis of the capabilities of mod-

ern commodity hardware in processing video streams, discussing potential deployment challenges. Finally, Section 4 concludes this paper.

2. Deep Video Stream Information Analysis and Retrieval

2.1 Deep Semantic-based Media Analysis

Several DL tools for media analysis have been developed in recent years, overcoming most of the limitations of previously used methods, ranging from generic object detection approaches and segmentation methods [55, 20] to specialized human-centric methods for extracting various behavioral analytics [53]. Indeed, a common task that arises in media monitoring applications is object and logo detection. The former can be easily tackled with existing object detection approaches [23, 34]. Building upon these methods, logo detection and recognition methods have been also developed [5, 39, 21]. Such approaches enable advertisers to monitor the effectiveness of their campaigns in various media. These approaches are typically highly accurate, yet they require a tedious manual annotation and retraining each time a new object category and/or logo much be added for detection. This can significantly hinder their application in production systems since it significantly increases the deployment cost. Indeed, as we further explain in Section 2.2, employing a slightly different retrieval-based pipeline allow us to effectively overcome these limitations.

It is also worth noting that in some cases we are not interested in localizing an object, but just answering whether it is present in a given frame. Indeed, in many media monitoring applications, we are interested in measuring the time a concept appears in a given video stream, instead of precisely localizing it inside each frame. A typical example of such an application is measuring total product placement time. In such cases, we can apply concept-based recognition approaches [25, 19, 45]. These approaches allow for extracting such metadata faster and - in some cases - with higher precision since they are capable of processing higher dimensional inputs [48].

Apart from object detection, powerful person and face detection methods have been developed, allowing for detecting humans with high accuracy [11, 17, 23]. These methods, despite being significantly faster than previous proposal-based detection methods, can still require a significant amount of resources. At the same time, it is worth noting that these approaches, despite being very accurate, usually work on relatively low-resolution inputs. As a result, when there is a need for detecting very small faces, specialized methods are typically employed, e.g., using multi-scale detection [44].

Moreover, DL also provided powerful tools for face recognition, which can effectively scale into datasets with millions of people [52]. However, it should be noted that all of these methods expect a cropped version of a face to be recognized. As a result, face detection is usually a preprocessing step for every face recognition application, increasing the complexity of DL pipelines for face recognition. Face alignment steps can be also employed, which can further improve the accuracy. Such steps have been incorporated into recent face detection pipelines, such as [8], further accelerating the resulting pipeline. Furthermore, closely related to face recognition approaches are methods that focus on re-identifying persons that appear in different frames, e.g., in a mall that has different cameras. Several powerful DL formulations for person re-identification have also been developed [60]. It is worth noting that these approaches typically do not only focus on facial features but take into account the whole appearance of a person, e.g,. clothes, haircut, etc. Therefore, most of these approaches focus on re-identifying a subject within very small time horizons.

Finally, several other DL models for various visual analytics have been proposed. These range from pose estimation [42] and emotion recognition approaches [26] to activity recognition, both from still images and video [53]. All these tools, provide unique opportunities for automated extraction of analytics regarding the behavior of humans in various settings. At the same time, these capabilities have also raised significant issues regarding the way such data are processed. However, it is worth noting that DL for edge deployment can often allow such analytics to be extracted in an anonymized fashion [3], without transferring or storing any data to the cloud, fully ensuring the privacy of the end-users.

2.2 Deep Information Retrieval

Employing retrieval-based approaches is also especially useful in various video stream analysis applications, ranging from retrieving similar videos or images from large-scale databases [2] to efficiently handling few-shot learning tasks [43]. Indeed, following the successful application of DL for tackling a wide spectrum of visual recognition problems, a plethora of recent works also resort to DL to tackle information retrieval, and especially image retrieval, accomplishing superior performance over previous shallow approaches [7, 10, 29]. Generally, deep image retrieval methods fall into two broad categories: unsupervised methods that employ reconstruction-based objectives [16], and supervised ones, that, typically accomplish superior performance over the former ones, by learning discriminative representations through supervised objectives [37, 50]. However, employing highly discriminative objectives can often

lead to overfitting the training domain, failing to generalize to unseen, yet related domains. To this end, methods that can exploit the geometric structure of the data in an unsupervised fashion, as well as the user's feedback using relevance feedback have been proposed [47, 46].

Such information retrieval approaches can enable to perform various video stream information analysis tasks very effectively, especially when the needs of the end-users can quickly shift. For example, consider the task of logo detection where we can decouple the detection and recognition tasks. In this case, we can first train a generic object logo detector to detect any entity on a frame that can potentially look like a logo, which is a task that can be easily handled given the existence of large-scale datasets that can be used to develop detectors that can detect even unseen logos [51]. Then, we can tackle the problem of logo recognition as a separate step, as in [4], either by employing few-shot learning or any information retrieval approach. Such setup enables for easily augmenting the logos that can be recognized by the system just by adding a few of them into the logo database without the need for re-training the whole DL detector, providing an effective solution for continual learning [24] with minimal effort for the end-user.

2.3 Optimization DL models for large-scale deployment

Despite the achievements of DL in the aforementioned areas, the computational and energy requirements of them can be often a limiting factor when deploying them in large-scale applications [40], as also mentioned in Section 1. This led to the development of a wide range of methods for developing lightweight, yet almost equally effective models. Some methods focused on reducing the complexity of operations, i.e., both memory and computations, by reducing the number of bits used to store each parameter of the model, as well as their activations. Such methods are called *quantization* methods and have flourished in recent years [6, 9, 22], since most modern hardware can directly exploit the benefits provided by such approaches. It is worth noting that some dedicated DL accelerators, such as Edge TPUs [54], mostly operate on such quantized integer arithmetic, avoiding the need for performing floating-point operations as much as possible. Another approach that also focuses on removing redundant complexity from DL models is neural network pruning [1, 14, 56]. These approaches focus on removing unnecessary connections between neurons and/or whole neurons. Given that DL models are over-parametrized, such approaches can significantly reduce their size with only a very small impact on their accuracy. However, it is worth noting that in contrast with quantization that can be often very easily implemented in practice, since it is often supported by the underlying hardware, pruning can sometimes lead to architectures that do not always bring the anticipated speedup.

The aforementioned approaches allow us to either compress an already-trained network architecture or train it in a way that will eventually end up with fewer parameters/consume fewer resources. However, these approaches ignore the characteristics of modern DL accelerators, for which architectures with a similar number of parameters might perform differently. This has also fueled the interest in developing a lightweight architecture that will maximize the accuracy that we can obtain for the task at hand while being as much as possible hardware-friendly. Indeed, architectures such as MobileNets [15, 36], ShuffleNets [58] and EfficientNets [41] are predominately used when there is the need for fast, yet effective networks. Other approaches also focused on developing layers that can speed up the operation of a network, e.g., efficient pooling layers [31, 35]. Another line of work employs models with adaptive inference graphs [30, 49, 59], which allows for accelerating inference for easier samples and/or when not enough resources are available, reducing the footprint of DL models.

Usually compressing DL architectures and/or using such lightweight architecture negatively impacts the accuracy of the network (to a smaller or larger degree). Motivated by this behavior and exploiting the nature of DL models that tends to be heavily over-parameterized several methodologies for *distilling*/transferring the knowledge from a large and powerful network into a smaller and more lightweight one have been developed [13, 32, 57]. These methods exploit the additional information that can be extracted from the teacher model to improve the training process of the student, often leading to significant improvements in the accuracy of the student models.

Finally, apart from the aforementioned generic methods for accelerating DL models, a small number of methods tailored to specific applications have also been developed. For example, in [28] the problem of object detection is tackled as a detection and tracking problem, where a tracker is employed to accelerate re-detection, which can be very efficient when there are only small changes between successive frames. On the other hand, in [33] active perception approaches have been proposed which allows for training models that need to be less invariant to various poses, since the most appropriate one can be acquired, which in turn allows for faster inference in most situations.

3 Deployment Challenges

In this Section we provide experimental measurements to benchmark well-known architectures that can be used for various deep video stream analysis tasks. Since most of the recent DL approaches share common backbones, e.g., residual networks [12], that are then fine-tuned for the task

Table 1. Experimental evaluation and comparison between different architectures, models and resolutions that can be used for analyzing video streams on commodity GPUs. Memory footprint (MB) and energy footprint (Joules) refer to the amortized value per sample. The maximum batch size that can fit in the memory of GPU is also reported (Max. Batch).

Architecture	Memory Footprint	Energy Footprint	NVIDIA 3080 Ti		NVIDIA 2080 Ti	
	(MB)	(J)	Max. Batch	FPS	Max. Batch	FPS
Input Resolution: 360×640						
ResNet-18	209.2	0.5	49	566.4	49	421.0
ResNet-50	790.4	1.9	13	183.1	12	167.7
ResNet-101	1196.2	3.2	8	110.7	8	102.5
MobiletNet v2	483.7	1.1	20	382.9	20	379.5
ShuffleNet v2	94.15	0.2	106	1213.7	94	1096.3
Input Resolution: 720×1280						
ResNet-18	805.9	1.7	12	181.7	12	122.0
ResNet-50	3079.8	7.8	3	54.7	3	41.9
ResNet-101	4583.3	12.4	2	31.7	2	25.5
MobiletNet v2	1917.9	4.3	5	126.6	5	92.0
ShuffleNet v2	373.5	0.7	27	357.9	24	272.6
Input Resolution: 1080×1920						
ResNet-18	1783.7	4.2	5	69.6	5	51.4
ResNet-50	6918.6	23.1	1	19.5	1	14.9
MobiletNet v2	4302.9	10.5	2	44.4	2	33.9
ShuffleNet v2	840.76	1.7	12	147.8	10	120.4

at hand, this evaluation focuses on these backbones, which typically consume the largest share of resources. The experimental results are reported in Table 1. We have used two widely available GPUs that can be used for accelerating DL models, i.e., an NVIDIA 2080 Ti (11GB of VRAM, 250W TDP) and an NVIDIA RTX-3080 Ti (12GB of VRAM, 350W TDP). We are reporting the memory footprint required for inference of one frame, which provides an indication of the memory utilization of each model. Furthermore, we also report the number of frames that can be processed in parallel in one second, as well as the maximum number of frames that can fit in the VRAM of each card. These two-number provide an indication of the capacity of each card to handle parallel video streams. Moreover, given the increasing importance of energy consumption in DL, we also report average energy consumption per card (whole package, as reported by `nvidia-smi`) for processing one input frame. The aim of this evaluation was to estimate the efficiency of different networks on different architectures. The memory footprint (reported in MB) and energy footprint (reported in Joules) are calculated as the amortized value after setting the batch size to the maximum that can be supported (Max. Batch). These values were measured using an NVIDIA 3080 Ti GPU. Finally, it should be noted that memory and energy footprint might differ from the actual footprint measured at the device due to measurements limitations (e.g., ± 5 W accuracy for energy measurements).

Several interesting conclusions can be drawn from the results reported in Table 1. First, note that the employed architecture can have a profound impact on energy consumption (a ResNet-101 requires almost 16 times more energy compared to a ShuffleNet). The same is also true for the input resolution. Indeed, energy increases about 8-10 times as we increase the input resolution from 360×640 to 1080×1920. At the same time, note that memory footprint is becoming a limiting factor as the input resolution is increasing, highlighting a bottleneck point of modern GPU accelerators. Indeed, for a relatively small resolution we can process a significant number of frames, e.g., assuming 5 FPS per input video stream, we can process about 10 video streams using a ResNet-50 at 720p. However, when scaling to 1080p this number drops to just 3-4 video steams, while the GPU memory is enough only for performing inference for one frame. Going from a previous generation card (NVIDIA 2080 Ti) into a more recent one (3080 Ti) increases the inference speed, but memory restrictions still remain a significant limitation when there is the need for high-resolution processing.

4 Conclusions

In this paper we provided a brief overview of the current state-of-the-art in DL tools that can be used for deep video stream information analysis and retrieval, discussing critical limitations, as well as modern methods and tools for overcoming them. Furthermore, we also include experimental results discussing deployment challenges, ranging from speed to energy consumption, demonstrating the capabilities of readily available commodity hardware in processing video streams for selected DL models.

Acknowledgements: This research was funded by the project "SEMANTIC ANNOTATION AND METADATA ENRICHMENT OF OPEN VIDEO STREAMS USING DEEP LEARNING" (Project code: KMP6-0079092) that was implemented under the framework of the Action "Investment Plans of Innovation" of the Operational Program "Central Macedonia 2014 2020", that is co-funded by the European Regional Development Fund and Greece.

References

[1] S. Anwar, K. Hwang, and W. Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems*, 13(3):1–18, 2017.

[2] A. Araujo and B. Girod. Large-scale video retrieval using image queries. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6):1406–1420, 2017.

[3] M. N. Asghar, M. S. Ansari, N. Kanwal, B. Lee, M. Herbst, and Y. Qiao. Deep learning based effective identification of eu-gdpr compliant privacy safeguards in surveillance videos. In *IEEE Intl. Conf. Dependable, Autonomic and Secure Computing*, pages 819–824, 2021.

[4] A. K. Bhunia, A. K. Bhunia, S. Ghose, A. Das, P. P. Roy, and U. Pal. A deep one-shot network for query-based logo retrieval. *Pattern Recognition*, 96:106965, 2019.

[5] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini. Deep learning for logo recognition. *Neurocomputing*, 245:23–30, 2017.

[6] Z. Cai, X. He, J. Sun, and N. Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 5918–5926, 2017.

[7] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. Lew. Deep image retrieval: A survey. *ArXiv*, 2021.

[8] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.

[9] N. Floropoulos and A. Tefas. Complete vector quantization of feedforward neural networks. *Neurocomputing*, 367:55–63, 2019.

[10] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067, 2020.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proc. IEEE Intl. Conf. Computer Vision*, pages 2961–2969, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] G. Hinton, O. Vinyals, J. Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[14] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124, 2021.

[15] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In *Proc. IEEE/CVF Intl. Conf. Computer Vision*, pages 1314–1324, 2019.

[16] A. Krizhevsky and G. E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, volume 1, page 2, 2011.

[17] A. Kumar, A. Kaur, and M. Kumar. Face detection techniques: a review. *Artificial Intelligence Review*, 52(2):927–948, 2019.

[18] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] H. Li, L. Liu, F. Sun, Y. Bao, and C. Liu. Multi-level feature representations for video semantic concept detection. *Neurocomputing*, 172:64–70, 2016.

[20] Y. Li, L. Wen, M.-C. Chang, and S. Lyu. Graph-to-graph energy minimization for video object segmentation. In *Proc. Intl. Conf. Advanced Video and Signal Based Surveillance*, pages 1–8, 2019.

[21] Y. Liao, X. Lu, C. Zhang, Y. Wang, and Z. Tang. Mutual enhancement for detection of multiple logos in sports videos. In *Proc. IEEE Intl. Conf. Computer Vision*, pages 4846–4855, 2017.

[22] X. Lin, C. Zhao, and W. Pan. Towards accurate binary convolutional neural network. *Proc. Advances in Neural Information Processing Systems*, 30, 2017.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proc. European Conf. Computer Vision*, pages 21–37, 2016.

[24] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. *Proc. Advances in Neural Information Processing Systems*, 30, 2017.

[25] F. Markatopoulou, V. Mezaris, and I. Patras. Deep multi-task learning with label correlation constraint for video concept detection. In *Proc. ACM Intl. Conf. Multimedia*, pages 501–505, 2016.

[26] W. Mellouk and W. Handouzi. Facial emotion recognition using deep learning: review and insights. *Procedia Computer Science*, 175:689–694, 2020.

[27] B. Mitra, N. Craswell, et al. *An Introduction to Neural Information Retrieval*. Now Foundations and Trends Boston, MA, 2018.

[28] P. Nousi, D. Triantafyllidou, A. Tefas, and I. Pitas. Re-identification framework for long term visual object tracking based on object detection and classification. *Signal Processing: Image Communication*, 88:115969, 2020.

[29] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proc. ACM on Conf. Information and Knowledge Management*, pages 257–266, 2017.

[30] N. Passalis, J. Raitoharju, A. Tefas, and M. Gabbouj. Efficient adaptive inference for deep convolutional neural networks using hierarchical early exits. *Pattern Recognition*, 105:107346, 2020.

[31] N. Passalis and A. Tefas. Learning bag-of-features pooling for deep convolutional neural networks. In *Proc. IEEE Intl. Conf. Computer Vision*, pages 5755–5763, 2017.

[32] N. Passalis and A. Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proc. European Conf. Computer Vision*, pages 268–284, 2018.

[33] N. Passalis and A. Tefas. Leveraging active perception for improving embedding-based deep face recognition. In *Proc. IEEE Intl. Workshop on Multimedia Signal Processing*, pages 1–6, 2020.

[34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[35] O. Saha, A. Kusupati, H. V. Simhadri, M. Varma, and P. Jain. Rnnpool: efficient non-linear pooling for ram constrained inference. *Proc. Advances in Neural Information Processing Systems*, 33:20473–20484, 2020.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[37] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[38] V. Sharma, M. Gupta, A. Kumar, and D. Mishra. Video processing using deep learning techniques: A systematic literature review. *IEEE Access*, 2021.

[39] H. Su, S. Gong, and X. Zhu. Scalable deep learning logo detection. *arXiv preprint arXiv:1803.11417*, 2018.

[40] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE*, 105(12):2295–2329, 2017.

[41] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. Intl. Conf. Machine Learning*, pages 6105–6114, 2019.

[42] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[43] E. Triantafillou, R. Zemel, and R. Urtasun. Few-shot learning through an information retrieval lens. *Proc. Advances in Neural Information Processing Systems*, 30, 2017.

[44] D. Triantafyllidou, P. Nousi, and A. Tefas. Fast deep convolutional face detection in the wild exploiting hard sample mining. *Big Data Research*, 11:65–76, 2018.

[45] M. Tzelepi and A. Tefas. Human crowd detection for drone flight safety using convolutional neural networks. In *Proc. European Signal Processing Conference*, pages 743–747.

[46] M. Tzelepi and A. Tefas. Deep convolutional image retrieval: A general framework. *Signal Processing: Image Communication*, 63:30–43, 2018.

[47] M. Tzelepi and A. Tefas. Deep convolutional learning for content based image retrieval. *Neurocomputing*, 275:2467–2478, 2018.

[48] M. Tzelepi and A. Tefas. Improving the performance of lightweight cnns for binary classification using quadratic mutual information regularization. *Pattern Recognition*, 106:107407, 2020.

[49] A. Veit and S. Belongie. Convolutional networks with adaptive inference graphs. In *Proc. European Conf. Computer Vision*, pages 3–18, 2018.

[50] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proc. ACM Intl. Conf. Multimedia*, pages 157–166, 2014.

[51] J. Wang, W. Min, S. Hou, S. Ma, Y. Zheng, and S. Jiang. Logodet-3k: A large-scale image dataset for logo detection. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(1):1–19, 2022.

[52] M. Wang and W. Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021.

[53] G. Yao, T. Lei, and J. Zhong. A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118:14–22, 2019.

[54] A. Yazdanbakhsh, K. Seshadri, B. Akin, J. Laudon, and R. Narayanaswami. An evaluation of edge tpu accelerators for convolutional neural networks. *arXiv preprint arXiv:2102.10423*, 2021.

[55] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proc. European Conf. Computer Vision*, pages 325–341, 2018.

[56] R. Yu, A. Li, C.-F. Chen, J.-H. Lai, V. I. Morariu, X. Han, M. Gao, C.-Y. Lin, and L. S. Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 9194–9203, 2018.

[57] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proc. IEEE/CVF Intl. Conf. Computer Vision*, pages 3713–3722, 2019.

[58] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[59] Z. Zhao, K. M. Barijough, and A. Gerstlauer. Deepthings: Distributed adaptive deep learning inference on resource-constrained iot edge clusters. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2348–2359, 2018.

[60] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Proc. IEEE Intl. Conf. Computer Vision*, pages 1116–1124, 2015.