EXPERIMENTALLY DEFINED CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE VARIANTS FOR NON-TEMPORAL REAL-TIME FIRE DETECTION

Andrew J. Dunnings[1], *Toby P. Breckon*[1,2]
Department of {Computer Science[1] | Engineering[2]}, Durham University, UK.

ABSTRACT

In this work we investigate the automatic detection of fire pixel regions in video (or still) imagery within real-time bounds without reliance on temporal scene information. As an extension to prior work in the field, we consider the performance of experimentally defined, reduced complexity deep convolutional neural network architectures for this task. Contrary to contemporary trends in the field, our work illustrates maximal accuracy of 0.93 for whole image binary fire detection, with 0.89 accuracy within our superpixel localization framework can be achieved, via a network architecture of signficantly reduced complexity. These reduced architectures additionally offer a 3-4 fold increase in computational performance offering up to 17 fps processing on contemporary hardware independent of temporal information. We show the relative performance achieved against prior work using benchmark datasets to illustrate maximally robust real-time fire region detection.

Index Terms— simplified CNN, fire detection, real-time, non-temporal, non-stationary visual fire detection

1. INTRODUCTION

A number of factors have driven forward the increased need for fire (or flame) detection within video sequences for deployment in a wide variety of automatic monitoring tasks. The increasing prevalence of industrial, public space and general environment monitoring using security-driven CCTV video systems has given rise to the consideration of these systems as secondary sources of initial fire detection (in addition to traditional smoke/heat based systems). Furthermore, the on-going consideration of remote vehicles for fire detection and monitoring tasks [1, 2, 3] adds further to the demand for autonomous fire detection from such platforms. In the latter case, attention turns not only to the detection of fire itself but also its internal geography of the fire and temporal development [4].

Traditional approaches in this area concentrate either on the use of a purely colour based approach [5, 6, 7, 8, 9, 4] or a combination of colour and high-order temporal information [10, 11, 12, 13]. Early work emanated from the colour-threshold approach of [5] which was extended with the basic consideration of motion by [10]. Later work considered the temporal variation (flame flicker) of fire imagery within the Fourier domain [11] with further studies formulating a Hidden Markov Model problem [12]. More recently work considering the temporal aspect of the problem has investigated time-derivatives over the image [13]. Although flame flicker is generally not sinusoidal or periodic under all conditions, a frequency of 10Hz has been observed in generalised observational studies [14]. As such, [15] considered the use of the wavelet transform as a temporal feature. In later applications [7], we still see the basic approaches of [10] underlying colour-driven approaches although more sophisticated colour models based on a derivative of background segmentation [9] and consideration of alternative colour spaces [8] are proposed. In general these works report ~98-99% (true positive) detection at 10-40 frames per second (fps) on relatively small image sizes (CIF or similar) [9, 8].

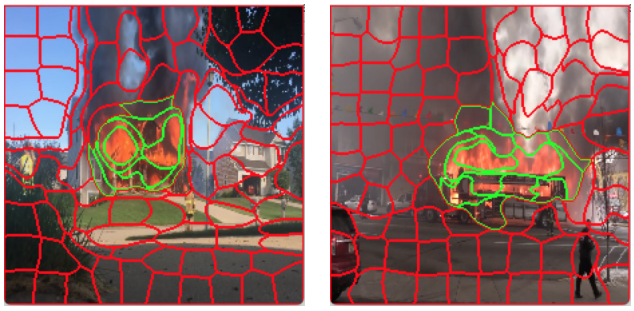

Fig. 1. Example fire detection and localization.

More recent work has considered machine learning based classification approaches to the fire detection problem [3, 16, 17]. The work of [3] considers a colour-driven approach utilising temporal shape features as an input to a shallow neural network and similarly the work of [16] utilises wavelet co-efficients as an input to a SVM classifier. Chenebert et al. [17] consider the use of a non-temporal approach with the combined use of colour-texture feature descriptors as an input to decision tree or shallow neural network classification (80-90% mean true positive detection, 7-8% false positive). Other recent approaches consider the use shape-invariant features [18] or simple patches [19] within varying machine learning approaches. However, the majority of recent work is temporally dependent considering a range of dynamic features [20] and motion characteristics [21, 22] between consecutive video frames with the most recent work of [22] considering convolutional neural networks (CNN) for fire detection within this context.

Here, by contrast to previous classifier-driven work [3, 16, 4, 21, 20, 22], we instead consider a non-temporal classification model for fire detection following the theme non-temporal fire detection championed by Chenebert et al. [17] and further supporting by the non-stationary camera visual fire detection challenge posed by Steffans et al. [23]. Non-temporal detection models are highly suited to the non-

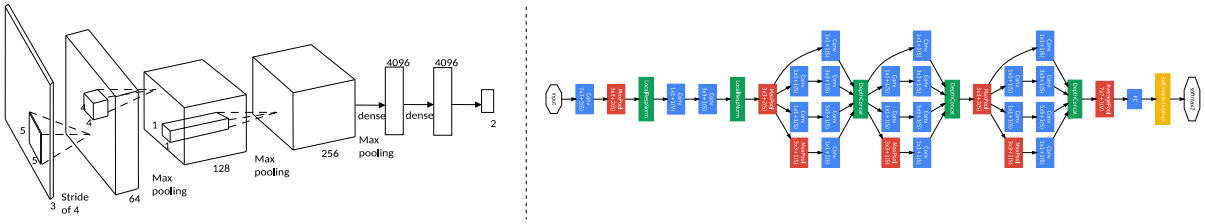

Fig. 2. Reduced complexity CNN architectures for FireNet (left) and InceptionV1-OnFire (right) optimized for fire detection.

stationary fire detection scenario posed by the future use of autonomous systems in a fire fighting context [23]. Within this work we show that comparable fire detection results are achievable to the recent temporally dependent work of [21, 20, 22], both exceeding the prior non-temporal approach of Chenebert et al. [17] and within significantly lower CNN model complexity than the recent work of [22]. Our reduced complexity network architectures are experimentally defined as architectural subsets of seminal CNN architectures offering maximal performance for the fire detection task. Furthermore, we extend this concept to incorporate in-frame localization via the use of superpixels [24] and benchmark comparison using the fire non-stationary (moving camera) visual fire detection dataset released under [23].

2. APPROACH

Our approach centres around the development of low-complexity CNN architectural variants (Section 2.1) operating on single image inputs (non-temporal) experimentally optimized for the fire detection task (Section 2.2). This is then expanded into a superpixel based localization approach (Section 2.3) to offer a complete detection solution.

2.1. Reference CNN Architectures

We consider several candidate architectures, with reference to general object recognition performance within [25], to cover varying contemporary CNN design principles [26] that can then form the basis for our reduced complexity CNN approach.

AlexNet [27] represents the seminal CNN architecture comprising of 8 layers. Initially, a convolutional layer with a kernel size of 11 is followed by another convolutional layer of kernel size 5. The output of each of these layers is followed by a max pooling layer and local response normalization. Three more convolutional layers then follow, each having a kernel size of 3, and the third is followed by a max pooling layer and local response normalization. Finally, three fully connected layers are stacked to produce the classification output.

VGG-16 [28] is a network architecture based on the principle of prioritizing simplicity and depth over complexity – all convolutional layers have a kernel size of 3, and the network has a depth of 16 layers. This model consists of groups of convolutional layers, and each group is followed by a max pooling layer. The first group consists of two convolutional layers, each with 64 filters, and is followed by a group of two convolutional layers with 128 filters each. Subsequently, a group of three layers with 256 filters each, and another two groups of three layers with 512 filters each feed into three fully connected layers which produce the output. Here we implement the 13-layer variant of this network by removing one layer from each of the final three groups of convolutional layers (denoted VGG-13).

InceptionV1 ([29], GoogLeNet) is a network architecture composed almost entirely of a single repeating inception module element consisting of four parallel strands of computation, each containing different layers. The theory behind this choice is that rather than having to choose between convolutional filter parameters at each stage in the network, multiple different filters can be applied in parallel and their outputs concatenated. Different sized filters may be better at classifying certain inputs, so by applying many filters in parallel the network will be more robust. The four strands of computation are composed of convolutions of kernel sizes 1×1, 3×3, and 5×5, as well as a 3×3 max pooling layer. 1×1 convolutions are included in each strand to provide a dimension reduction – ensuring that the number of outputs does not increase from stage to stage, which would drastically decrease training speed. The InceptionV1 architecture offers a contrasting 22 layer deep network architecture to AlexNet (8 layers), offering superior benchmark performance [29], whilst having 12 times fewer parameters through modularization that make use of 9 inception modules in its standard configuration.

2.2. Simplified CNN Architectures

Informed by the relative performance of the three representative CNN architectures (AlexNet, VGG-13, InceptionV1) on the fire detection task (Table 1, upper), an experimental assessment of the marginally better performing AlexNet and InceptionV1 architectures is performed.

Our experimental approach systematically investigated variations in architectural configuration of each network against overall performance (statistical accuracy) on the fire image classification task. Performance was measured using the same evaluation parameters set out in Section 3 with network training performed on 25% of our fire detection training dataset and evaluated upon the same test dataset.

For AlexNet we consider six variations to the architectural configuration by removing layers from the original architecture, denoted by C1-C6 as follows: C1 removed layer 3 only, C2 removed layers $3, 4$, C3 removed layers $3, 4, 5$, C4 removed layers 6 only, C5 removed layers $3, 4, 6$ and C6 re-

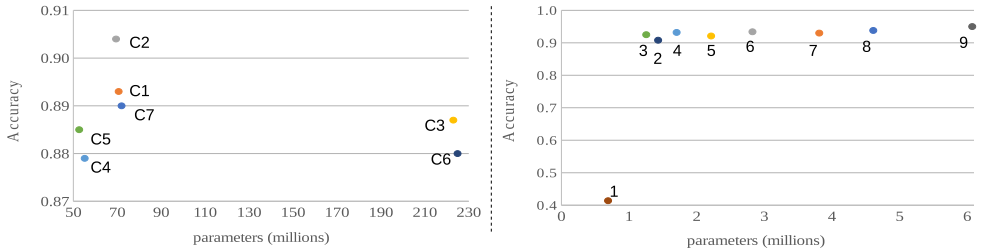

Fig. 3. Fire detection performance for variations of the AlexNet architecture (left) and InceptionV1 architecture (right).

moved layer 2 only. The results in terms of statistical accuracy for fire detection plotted against the number of parameters present in the resulting network model are shown in Figure 3 (left) where C7 represents the performance of the original AlexNet architecture [27].

For the InceptionV1 architecture we consider eight variations to the architectural configuration by removing up to 8 inception modules from the original configuration of 9 present [29]. The results in terms of statistical accuracy for fire detection plotted against the number of parameters present in the resulting model are shown in Figure 3 (right) where label $i \in \{1..8\}$ represents the resulting network model with only i inception modules present and $i = 9$ represents the performance of the original InceptionV1 architecture [29].

From the results shown in Figure 3 (left) we can see that configuration C2 improves upon the accuracy of all other architectural variations whilst containing significantly less parameters than several other configurations, including the original AlexNet architecture. Similarly, from the results shown in Figure 3 (right) we can see that accuracy tends to slightly decrease as the number of inception modules decreases, whereas the number of parameters decreases significantly. The exception to this variation is using only one inception module, for which performance is significantly reduced. An architecture containing only three inception modules is the variation with the fewest parameters which retains performance in the highest band (Figure 3, right).

Overall from our experimentation on this subset of the main task (i.e. 25% training data), we can observe both explicit over-fitting within these original high-complexity CNN architectures such as the performance of reduced CNN C2 vs. original AlexNet architecture C7 (Figure 3, left) and also the potential for over-fitting where significantly increased architectural complexity within a InceptionV1 modular paradigm offers only marginal performance gains (Figure 3, right). Based on these findings, we propose two novel reduced complexity CNN architectures targeted towards performance on the fire detection task (illustrated in Figure 2).

FireNet is based on our C2 AlexNet configuration such that it contains only three convolutional layers of sizes 64, 128, and 256, with kernel filter sizes 5×5, 4×4, and 1×1 respectively. Each convolutional layer is followed by a max pooling layer with a kernel size 3×3 and local response normalization. This set of convolutional layers are followed by

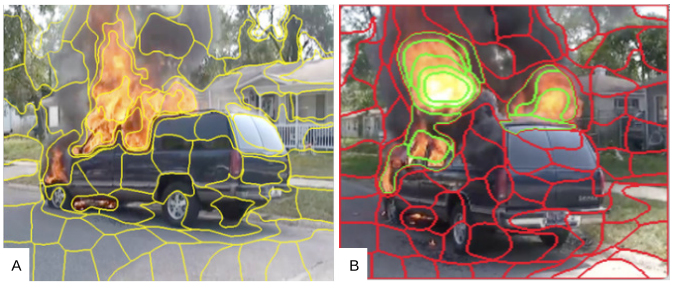

Fig. 4. Exemplar superpixel based fire region localization (A) and subsequent CNN based classification (B).

two fully connected layers, each with 4096 incoming connections and using $tanh()$ activation. Dropout of 0.5 is applied across these two fully connected layers during training to offset residual over-fitting. Finally we have a fully connected layer with 2 incoming connections and soft-max activation output. The architecture of FireNet is illustrated in Figure 2 (left) following the illustrative style of the original AlexNet work to aid comparison.

InceptionV1-OnFire is based on the use of a reduced InceptionV1 architecture only with three consecutive inception modules. Each individual module follows the same definition as the original work [29], using these first three in the same interconnected format as in the full InceptionV1 architecture. As shown in Figure 2 (right), following the illustrative style of the original InceptionV1 work to aid comparison, the same unchanged configuration of pre-process and post-process layers are used around this three module set.

2.3. Superpixel Localization

In contrast to earlier work [17, 8] that largely relies on colour-based initial localization, we instead adopt the use of superpixel regions [24]. Superpixel based techniques over-segment an image into perceptually meaningful regions which are similar in colour and texture (Figure 4). Specifically we use simple linear iterative clustering (SLIC) [24], which essentially adapts the k-means clustering to reduced spatial dimensions, for computational efficiency. An example of superpixel based localization for fire detection is shown in Figure 4A with classification akin to [30, 31] via CNN (Figure 4B).

3. EVALUATION

For the comparison of the simplified CNN architectures outlined we consider the True Positive Rate (TPR) and False Positive Rate (FPR) together with the F-score (F), Precision (P)

	TPR	FPR	F	P	A
AlexNet	0.91	**0.07**	0.93	0.95	0.92
InceptionV1	0.96	0.09	**0.95**	**0.94**	0.93
VGG-13	0.93	0.11	0.93	0.92	0.91
FireNet	0.92	0.09	0.93	0.93	0.92
InceptionV1-OnFire	**0.96**	0.10	0.94	0.93	**0.93**

Table 1. Statistical performance - full-frame fire detection.

	C ($\times 10^6$)	A (%)	A:C	fps
Alexnet	71.9	91.7	1.3	4.0
FireNet	68.3	91.5	1.3	**17.0**
InceptionV1	6.1	**93.4**	15.4	2.6
InceptionV1-OnFire	**1.2**	**93.4**	**77.9**	8.4
Chenebert et al. [17]	-	-	-	0.16

Table 2. Statistical results - size, accuracy and speed (fps).

and accuracy (A) statistics in addition to comparison against the state of the art in non-temporal fire detection [17]. We address two problems for the purposes of evaluation:- (a) full-frame binary fire detection (i.e. fire present in the image as whole - *yes/no?*) and (b) superpixel based fire region localization against ground truth in-frame annotation [23].

CNN training and evaluation was performed using fire image data compiled from Chenebert et al. [17] (75,683 images) and also the established visual fire detection evaluation dataset of Steffens et al. [23] (20593 images) in addition to material from public video sources (youtube.com: 269,426 images) to give a wide variety of environments, fires and non-fire examples (total dataset: 365,702 images). From this dataset a training set of 23,408 images was extracted for training and testing a full-frame binary fire detection problem (70:30 data split) with a secondary validation set of 2931 images used for statistical evaluation. Training is from random initialisation using stochastic gradient descent with a momentum of 0.9, a learning rate of 0.001, a batch size of 64 and categorical cross-entropy loss. All networks are trained using a a Nvidia Titan X GPU via TensorFlow (1.1 + TFLearn 0.3).

From the results presented in Table 1, addressing the full-frame binary fire detection problem, we can see that the InceptionV1-OnFire architecture matches the maximal performance of its larger parent network InceptionV1 (0.93 accuracy / 0.96 TPR, within 1% on other metrics). Furthermore, we can see a similar performance relationship between the FireNet architecture and its AlexNet parent.

Computational performance at run-time was performed using at average of 100 image frames of 608×360 RGB colour video on a Intel Core i5 2.7GHz CPU and 8GB of RAM. The resulting frames per second (fps) together with a measure of architecture complexity (parameter complexity, C), percentage accuracy (A) and ratio $A : C$ are shown in Table 2. From the results presented in Table 2,we observe significant run-time performance gains for the reduced complexity FireNet and InceptionV1-OnFire architectures compared to their parent architectures. Whilst FireNet provides a maximal 17 fps throughput, it is notable that InceptionV1-OnFire provides the maximal accuracy to complexity ratio. Whilst the accuracy of FireNet is only slightly worse than that of AlexNet, it can perform a classification 4.2$\times$ times faster. Similarly InceptionV1-OnFire matches the accuracy of InceptionV1 but can perform a classification 3.3$\times$ faster.

Detection (full-frame)	TPR	FPR	F	P	A
Chenebert et al. [17]	**0.99**	0.28	**0.92**	0.86	**0.89**
InceptionV1-OnFire	0.92	**0.17**	0.90	**0.88**	**0.89**

Localization (pixel region)	TPR	F	P	S
Chenebert et al. [17]	**0.98**	**0.90**	0.83	**0.80**
InceptionV1-OnFire	0.92	0.88	**0.84**	0.78

Table 3. Statistical results - localization).

To evaluate within the context of in-frame localization (Section 2.3), we utilise the ground truth annotation available from Steffens et al. [23] to label image superpixels for training, test and validation. The InceptionV1-OnFire architecture is trained over a set of 54,856 fire (positive) and 167,400 non-fire (negative) superpixel examples extracted from 90% of the image frames within [23]. Training is performed as per before with validation against the remaining 10% of frames comprising 1178 fire (positive) and 881 non-fire (negative) examples. The resulting contour from any fire detected superpixels is converted to a bounding rectangle and tested for intersection with the ground truth annotation (Similarity, S: correct if union over ground truth$>$0.5 as per [23]).

From the results presented in Table 3 (lower), we can see that the combined localization approach of superpixel region identification and localized InceptionV1-OnFire CNN classification performs marginally worse than the competing state of the art Chenebert et al. [17] but matching overall full-frame detection (Table 3, upper). However, as can be seen from Table 2, this prior work [17] has significantly worse computational throughput than any of the CNN approaches proposed here. Example detection and localization are shown in Figures 1 and 4B (fire = green, no-fire = red).

4. CONCLUSIONS

Overall we show that reduced complexity CNN, experimentally defined from leading architectures in the field, can achieve 0.93 accuracy for the binary classification task of fire detection. This significantly outperforms prior work in the field on non-temporal fire detection [17] at lower complexity than prior CNN based fire detection [22]. Furthermore, reduced complexity FireNet and InceptionV1-OnFire architectures offer classification accuracy within less than 1% of their more complex parent architectures at 3-4$\times$ of the speed (FireNet offering 17 fps). To these ends, we illustrate more generally a architectural reduction strategy for the experimentally driven complexity reduction of leading multi-class CNN architectures towards efficient, yet robust performance on simpler binary classification problems.

5. REFERENCES

[1] A. Bardshaw, "The UK security and fire fighting advanced robot project," in *IEE Coll. on Advanced Robotic Initiatives in the UK*, London, UK, 1991.

[2] J Martinezdedios, L Merino, F Caballero, A Ollero, and D Viegas, "Experimental results of automatic fire detection and monitoring with UAVs," *Forest Ecology and Management*, vol. 234, pp. S232–S232, Nov. 2006.

[3] D. Zhang, S. Han, J. Zhao, Z. Zhang, C. Qu, Y. Ke, and X. Chen, "Image based forest fire detection using dynamic characteristics with artificial neural networks," in *Proc. Int.Joint Conf. on Artificial Intell.*, 2009, pp. 290–293.

[4] F. Yuan, "An integrated fire detection and suppression system based on widely available video surveillance," *Mach. Vis. and Apps.*, vol. 21, pp. 941–948, 2010.

[5] G. Healey, D. Slater, T. Lin, B. Drda, and A.D. Goedeke, "A system for real-time fire detection," in *Proc. Int. Conf. Comp. Vis. and Pat. Rec.*, 1993, pp. 605–606.

[6] T. Chen, P. Wu, and Y. Chiou, "An early fire-detection method based on image processing," in *Proc. Int. Conf. on Image Proc.*, 2004, pp. 1707–1710.

[7] J. R. Martinez-de Dios, B. C. Arrue, A. Ollero, L. Merino, and F. Gómez-Rodríguez, "Computer vision techniques for forest fire perception," *Image Vision Comput.*, vol. 26, pp. 550–562, April 2008.

[8] T Celik and H Demirel, "Fire detection in video sequences using a generic color model," *Fire Safety J.*, vol. 44, no. 2, pp. 147–158, Feb. 2009.

[9] T Celik, H Demirel, H Ozkaramanli, and M Uyguroglu, "Fire detection using statistical color model in video sequences," *J. Vis. Comm. and Image Rep.*, vol. 18, no. 2, pp. 176–185, Apr. 2007.

[10] W. Phillips, M. Shah, and N. da Vitoria Lobo, "Flame recognition in video," *Pat. Rec. Letters*, vol. 23, no. 1-3, pp. 319–327, 2002.

[11] C. Liu and N. Ahuja, "Vision based fire detection," in *Proc. Int. Conf. on Pattern Recognition*, 2004, pp. 134–137.

[12] B.U. Toreyin, Y. Dedeoglu, and A.E. Cetin, "Flame detection in video using hidden Markov models," in *Proc. Int. Conf. on Image Proc.*, 2005, pp. II–1230.

[13] G Marbach, M Loepfe, and T Brupbacher, "An image processing technique for fire detection in video images," *Fire Safety J.*, vol. 41, no. 4, pp. 285–289, 2006.

[14] G. Q. James, *Principles of fire behavior*, Thomson Delmar Learning, 1997.

[15] B. Toreyin, Y. Dedeoglu, U. Gudukbay, and A. Cetin, "Computer vision based method for real-time fire and flame detection," *Patt. Rec. Letters*, vol. 27, no. 1, pp. 49–58, 2006.

[16] B. Ko, K. Cheong, and J. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Safety J.*, vol. 44, no. 3, pp. 322–329, Apr. 2009.

[17] A. Chenebert, T.P. Breckon, and A. Gaszczak, "A non-temporal texture driven approach to real-time fire detection," in *Proc. International Conference on Image Processing*. September 2011, pp. 1781–1784, IEEE.

[18] F. Yuan, "A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with adaboost for video smoke detection," *Pattern Recognition*, vol. 45, no. 12, pp. 4326–4336, 2012.

[19] J. Choi and J.Y. Choi, "Patch-based fire detection with online outlier learning," in *Proc. Int. Conf. Advanced Video and Signal Based Surveillance*. IEEE, 2015, pp. 1–6.

[20] R. Bohush and N. Brouka, "Smoke and flame detection in video sequences based on static and dynamic features," in *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2013*. IEEE, 2013, pp. 20–25.

[21] P. Morerio, L. Marcenaro, C.S. Regazzoni, and G. Gera, "Early fire and smoke detection based on colour features and motion analysis," in *Proc. Int. Conf. Image Processing*. IEEE, 2012, pp. 1041–1044.

[22] Y. Luo, L. Zhao, P. Liu, and D. Huang, "Fire smoke detection algorithm based on motion characteristic and convolutional neural networks," *Multimedia Tools and Applications*, pp. 1–18, 2017.

[23] C.R. Steffens, R.N. Rodrigues, and C.S. da Costa Botelho, "Non-stationary vfd evaluation kit: Dataset and metrics to fuel video-based fire detection development," pp. 135–151, 2016.

[24] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A.C. Berg, "Imagenet large scale visual recognition challenge," *Int. J. of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[27] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

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

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[30] J. Yan, Y. Yu, X. Zhu, X. Lei, and S.Z. Li, "Object detection by labeling superpixels," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015, pp. 5107–5116.

[31] W. Qin, J. Wu, F. Han, Y. Yuan, W. Zhao, B. Ibragimov, J. Gu, and L. Xing, "Superpixel-based and boundary-sensitive convolutional neural network for automated liver segmentation," *J. Physics in Medicine & Biology*, vol. 63, no. 9, pp. 095017, 2018.