ABSTRACT

Title of Thesis:	FAST–AT: FAST AUTOMATIC THUMBNAIL GENERATION USING DEEP NEURAL NETWORKS
	Seyed Abdulaziz Esmaeili, Master of Science, 2017
Thesis directed by:	Professor Larry S. Davis, Department of Computer Science

Fast-AT is an automatic thumbnail generation system based on deep neural networks. It is a fully-convolutional CNN, which learns specific filters for thumbnails of different sizes and aspect ratios. During inference, the appropriate filter is selected depending on the dimensions of the target thumbnail. Unlike most previous work, Fast-AT does not utilize saliency but addresses the problem directly. In addition, it eliminates the need to conduct region search on the saliency map. The model generalizes to thumbnails of different sizes including those with extreme aspect ratios and can generate thumbnails in real time. A data set of more than 70,000 thumbnail annotations was collected to train Fast-AT. We show competitive results in comparison to existing techniques.

FAST–AT: FAST AUTOMATIC THUMBNAIL GENERATION USING DEEP NEURAL NETWORKS

by

Seyed Abdulaziz Esmaeili

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2017

Advisory Committee:
Professor Larry S Davis, Chair
Professor David W. Jacobs,
Doctor Cornelia Fermuller

Table of Contents

Chapter 1: Introduction

1.1 Introduction to the thumbnail problem

Thumbnail images are reduced versions of original images that are meant to effectively portray the original image. Thumbnails facilitate the browsing of a large collection of images, make economic use of display size, and reduce the transmission time. Thumbnails are abundant on social media websites such as Facebook, Twitter, and Instagram. Figure 1 shows an example of an image and thumbnails of different sizes produced from that image. Figure 2 shows a typical example seen in many web pages where a large number of thumbnails are displayed together in a layout. It should be clear that displaying the original images instead of their thumbnails would occupy a much larger size and would make the browsing less efficient.

Figure 1. Original image on the left with thumbnails of different sizes on the right.

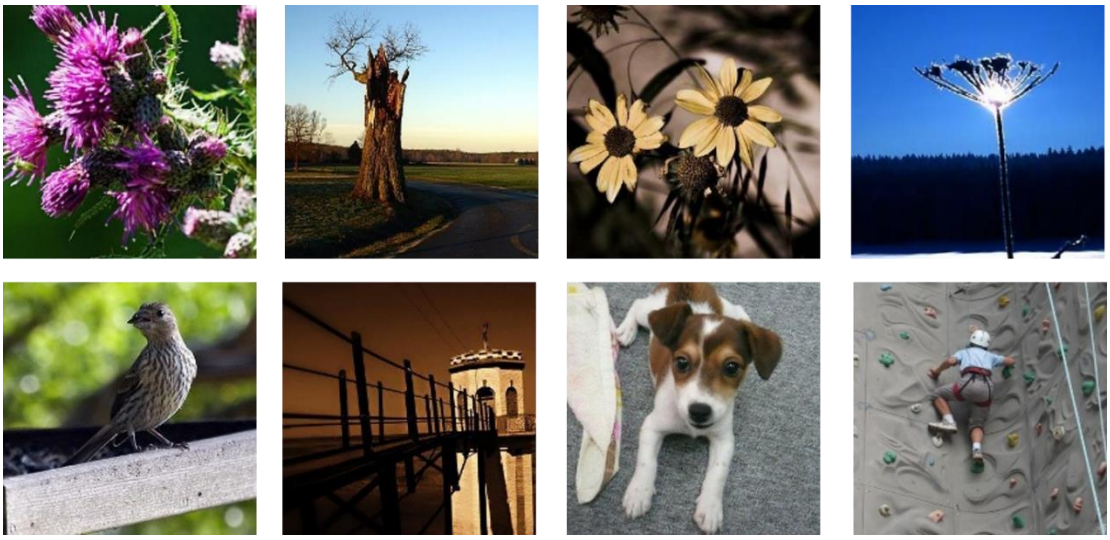

Figure 2. A collection of thumbnails shown in an array fashion, common to see in many websites.

It is clear that there is a significant connection between image retargeting and thumbnail generation. In image retargeting, a number of interesting and sophisticated methods have been introduced such as seam carving [25], non-homogenous warping [27], and multi-operator retargeting [26].The problem with these methods is that they are prone to generating pronounced artifacts. In addition, some of these methods assume that they will be used in a setting that is not fully automated, where the user may review the result of the retargeting method and possibly choose an alternate method to reduce the image size.

Because of the above reasons, automated thumbnail generation utilizes two main operations: cropping and scaling as shown in figure 3. Cropping and scaling are simple operations that are guaranteed not to produce artifacts. It is also interesting to note, that despite its simplicity, cropping ranked second in a user study that considered a number of retargeting methods [17]. Finally, the only thumbnail method

-we are aware of- that resorted to seam carving concluded that cropping achieved better performance in a user study [21].

Figure 3. Illustration of the thumbnail creation process, the original image is cropped and scaled down to the thumbnail size.

Therefore, given an image and a final thumbnail size, the production of the optimum thumbnail amounts to selecting the best crop (bounding box) in the original image and scaling it down to the final thumbnail size. The crop should accurately represent the original image and at the same time be easy to recognize. Extending the crop to the whole image, i.e. scaling the image down directly to the thumbnail size would produce a thumbnail that fully preserves the original content of the image but would make it harder to recognize, on the other hand a crop that encloses the main content of the image too tightly would produce a thumbnail that doesn't given an accurate representation of the original image.

1.2 Outline of this thesis:

In this thesis we focus on improving the automated generation of thumbnail images. In chapter 2, we will discuss previous methods used for thumbnail generation and their shortcomings. In chapter 3, we present our approach to thumbnail generation this includes: data set collection, brief review of deep neural networks and object

detection, architecture of our proposed deep learning model, and results. In chapter 4, we discuss the conclusion and future work.

Chapter 2: Related Work

2.1 Saliency Based Methods

As mentioned in chapter 1, automatic thumbnail generation amounts to selecting a crop that would be scaled down to the thumbnail size. That crop has to accurately represent the image while being easy to recognize at the same time. Because the notion of a representative region in the image is not well defined, much of the work in thumbnail generation utilized the saliency map as a heuristic indicator of the most representative regions in the image that should be enclosed in the crop. The crop size is limited by restricting the enclosed saliency to be below a certain threshold.

Suh et al. [20] were among the first to represent such a method. In their approach, the saliency map is first calculated, then region search is done to find a candidate set of crops that enclose a saliency above a certain threshold. The crop with the smallest area in the candidate set is then selected. Because this process is computationally expensive, a greedy search algorithm is used.

Sun et al. [21] takes the thumbnail size into account and enhances the saliency map by producing a scale aware saliency map that is then augmented with an objectness measure to finally produce a scale and object aware saliency map. A greedy search algorithm is used as done in [20] to select a crop. Another method they consider is a variant of seam carving [25]. However, the user study they conducted has shown that simple cropping is preferred.

A number of different methods were introduced to speed up the computation of the optimum crop. In [19], the search space in restricted to crops of specific sizes, in [3] the saliency map is binarized. Recently, an algorithm that has a linear complexity in the number of pixels was demonstrated [2]. It is also interesting to note that [2] represented an algorithm that can search for a crop with a specific aspect ratio. This is an important problem that was ignored in many methods, selecting crops with aspect ratios that differ from the final thumbnail aspect ratio results in thumbnails that look clearly deformed. This is shown in figure 4 which shows thumbnails produced with the code from [21].

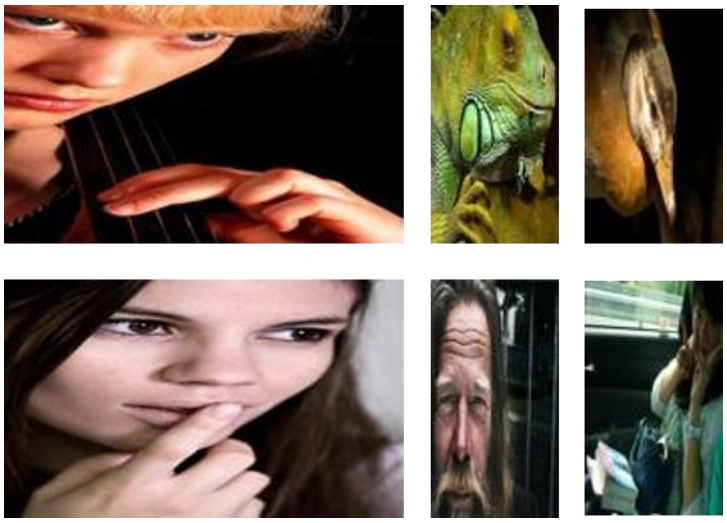

Figure 4. The above thumbnails were generated using the code from [21] which does not take the thumbnail aspect ratio into account. The produced thumbnails exhibit clear deformation.

However, it was noted that saliency can ignore the semantics of the scene and may lead to crops that miss important regions in the image. Therefore, some automatic

thumbnail generation and cropping methods made further considerations to produce better crops, such as selecting crops that encloses all of the detected faces[20] or first detecting the class of the image (landscape, close up, other) and then following a cropping algorithm that depends on the detected class [3].

Even if we assume that the saliency map is perfect for indicating representative regions in the image. Saliency based methods still have the following shortcomings. The saliency threshold should automatically adapt to the given image in a non-heuristic fashion. Since different threshold values lead to crops with different areas, [20, 21] choose a value for the threshold where the crop area gradient with respect to the threshold is very large. [2] suggests having the users adjust a threshold value because of the low complexity of their algorithm, however this is not an automated solution. Further, the algorithm presented in [2] for automated threshold selection considers only the case where there is no aspect ratio restriction and the derivation assumes that the image has high attention values that are spatially concentrated.

2.2 Image Aesthetics Based Methods

In aesthetic based image cropping, the crop which maximizes the quality of the visual appearance of the image is selected. Nishiyama et al. [15] generates a set of candidate crops from the original image and estimates the quality score of each candidate with a quality classifier.

In a similar line, Yan et al. [23] focus on producing crops that have a better overall composition without the distracting content of the original image. A data set of 1000 images that were cropped by expert photographers was collected and novel features were proposed to model the change in the image when a crop is selected.

2.3 Representativeness and Recognizability

Although aesthetic based methods produce crops that are visually pleasing and have been considered as thumbnail generation methods. They do not focus on the original problem of thumbnail generation which is a crop that leads to a thumbnail that is representative of the original image and recognizable.

The recent work in [9] attacks the problem directly and does not utilize saliency. A data set consisting of 600 images was collected, each image was cropped and scaled down to a thumbnail size of 160×120 by an expert photographer. The photographers were asked to produce thumbnails that give a good representation of the image while being easy to recognize at the same time. Feature engineering was done to produce a collection of features that faithfully model these two considerations. Support vector machines (SVMs) were trained with these features over the collected data set. Similar to [15, 23], at test time a set of candidate crops is generated by exhaustive sampling and the candidates are scored by the trained SVMs. The candidate with the highest score is selected and used to produce the final thumbnail.

However, a major shortcoming of their work, is that the system takes only a fixed thumbnail size of 160×120 and that it requires 60 seconds to generate a thumbnail for a single image.

Chapter 3: Deep Learning Based Approach for Automatic Thumbnail Generation

3.1 Motivation of the proposed solution

Saliency based thumbnail generation methods utilize saliency as a heuristic, and follow a two-step solution where the saliency map is first generated then region search is conducted. Unlike such methods, the optimum method should be similar to [9], focusing directly on producing thumbnails which are representable of the image and are easy to recognize and involving a one-step solution. It should also address the shortcomings of [9], i.e. it should generalize to thumbnails of different sizes and produce results in real time. Furthermore, in the recent years we have seen that deep learning based methods achieve a far better performance in high visual recognition tasks than methods based on SVMs trained over engineered features [4, 6, 7, 8, 11, 12]. Therefore, a deep learning based model should also be considered for thumbnail generation.

In this chapter we illustrate our proposed solution based on the above motivation. This covers data set collection, review of deep learning and object detection based using deep neural networks. We then present our proposed model, experiments conducted on the model and comparison to existing automatic thumbnail generation methods.

3.2 Data Set Collection

Deep learning based models have a much larger number of parameters which require larger data sets to prevent overfitting, therefore it is not enough to collect a data set of 600 images as was done in [9] to train the model. Furthermore, we note that the data set used in [9] which is the MIRFLICKR-25000 data set [10] mostly involves images that have a high quality and contain a single object in the foreground. More difficult data sets should be considered since many images received for automatic thumbnail generation systems in practice are not necessarily of high quality.

Therefore, we use images from the photo quality data set of [13]. This data consists of images that have both high and low quality and spans a number of categories such as man, animal, and landscape. For each image we select a thumbnail size. We choose thumbnail sizes in 3 groups: small thumbnail (from 32 to 64), medium thumbnails (from 64 to 128) and large thumbnails (from 100 to 200), the variation is for both width and height. This leads to an aspect ratio that varies from 0.5 to 2.

We use amazon mechanical turk (AMT) to annotate the data set. In the beginning, workers are shown examples of good and bad thumbnails that illustrate that the optimum thumbnail should enclose a representative region in the image while being easy to recognize at the same time. In the AMT interface, workers draw a bounding box (which represents the crop) on the original image. The bounding box has an aspect ratio that is equal to the thumbnail, workers can only move the box and scale it up or down. The selected bounding box is scaled down to the thumbnail size and

shown on the screen to the worker besides the image. Changes in the box lead to immediate changes in the resulting thumbnail which is displayed besides the original image. To make the interface more practical, the images were scaled down such that the height does not exceed 650 and the width does not exceed 800.The interface is illustrated in figure 5. At the end a data set of 70,048 thumbnail annotations over 28,064 images was collected.

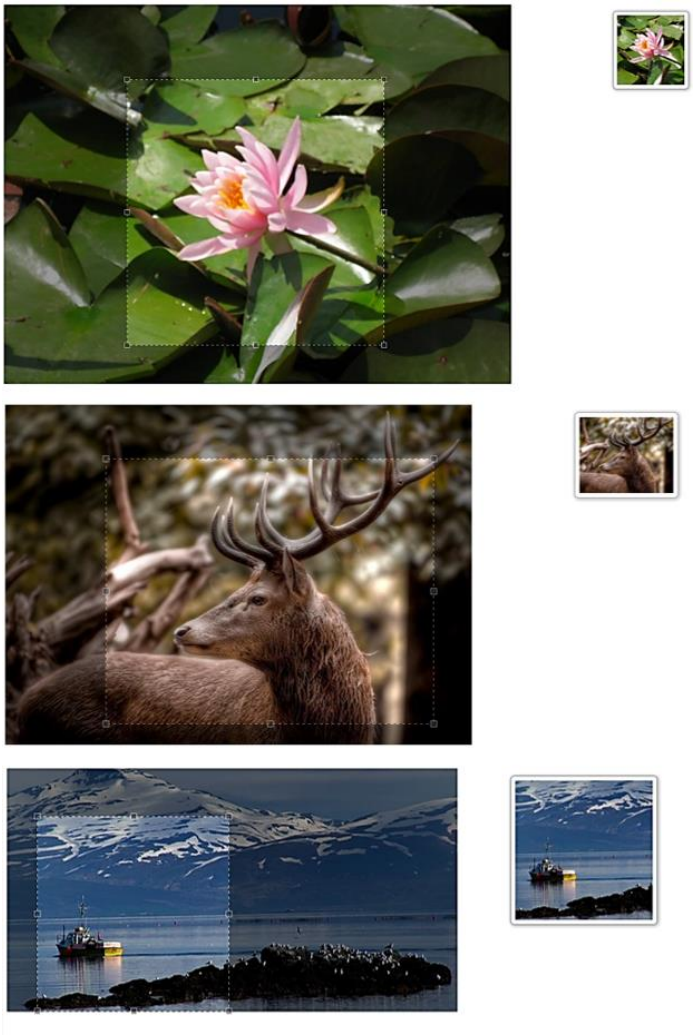

Figure 5. Illustration of the thumbnail generation interface. The original image is on the left, with a drawn bounding box, the resulting thumbnail is shown on the right at the same time.

3.3 Does the target thumbnail size matter?

Intuitively it is expected that smaller thumbnail sizes would require smaller crops since larger crops would be less recognizable when scaled down to the thumbnail size. To investigate this we plot the thumbnail area vs the average crop area. This would reveal whether this assumption is correct or not. As show in figure 6, the crop area does not tend to be smaller for smaller thumbnails. Hence, in the design of our system we do not take the thumbnail size into account, but only consider the aspect ratio, since crops of a different aspect ratio can lead to deformed thumbnails as shown in figure 4 when they are scaled down. We do however, consider a model which takes the thumbnail size into account to further verify the assumption.

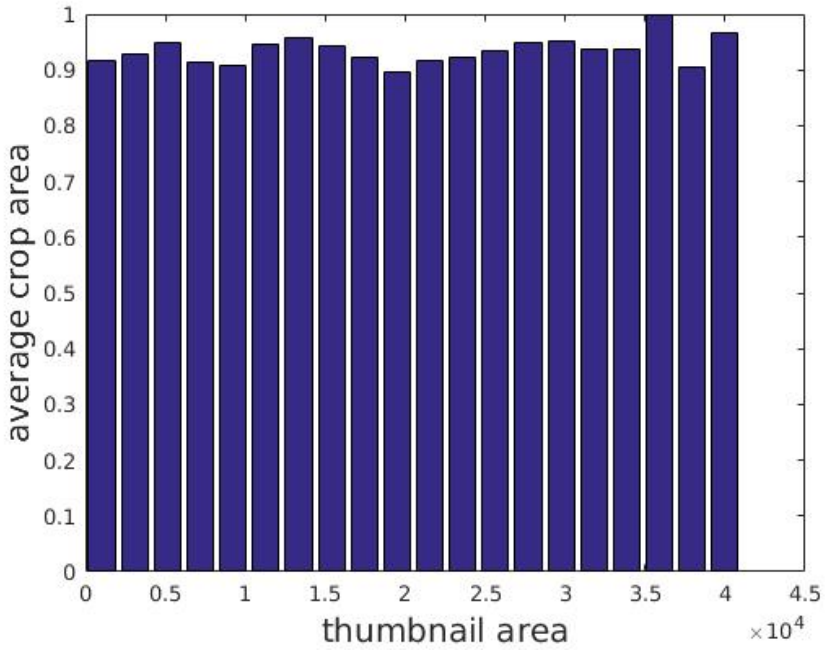

Figure 6. Plot of the thumbnail area vs average crop area.

3.3 Review of Deep Convolutional Neural Networks

Deep neural networks are high capacity machines that can be trained on raw data without requiring feature engineering. They can be represented in the form of a graph consisting of a number of layers. The first layer is the input layer which is usually the feature vector x. The preceding layer is called the hidden layer where each node outputs a dot product over the input vector plus a constant (bias) followed by the application of a nonlinear function. Specifically the output of a node is of the form: $(< w, x > + w_0)$, where x is the feature vector, w, and w_0 is the set of weights and bias associated with that node, respectively. σ is a nonlinear function, traditionally the sigmoid and the hyperbolic tangent functions where used, but it was found that the non-saturating ReLu [11] of the form $f(x) = \max(0, x)$ leads to faster convergence. The output layer follows a similar procedure to the hidden layer with the difference of changing the input vector x to the output vector of the hidden layer. It is possible for a network to have multiple hidden layers. The number of hidden layers plus the output layer is referred to as the depth of the network and mostly higher depth leads to better performance [33]. Deep learning models refers to neural networks with very large depth, as much as 101 or even more [8. 35].

All machine learning algorithms in a supervised setting, assume that there exists an annotated data set consisting of input feature vectors and output labels. A loss function is defined and then an optimization method is used to find the set of weights and parameters for this particular algorithm that will lead to a small loss. In the case of neural networks, the weight values are found through an algorithm called

backpropagation [29, 31]. Backpropagation is a simple algorithm that it based on successive application of the chain rule, until the gradient of the loss with respect to any given weight (or bias) is obtained (see[29] for details). The calculated gradient is used to adjust the weight in the right direction, according to the update rule $:= w - \eta \nabla_w l$, where η is the learning rate and $\nabla_w l$ is the gradient of the loss w.r.t to the weights. It is interesting to note that the function calculated by neural networks is non-convex in the weights and therefore the obtained solution could be a local minimum. However, in practice it has been found that solutions reached through backpropagation perform well. This has instigated theoretical research about this issue, such as [33].

Neural networks are not new to the community. They have gained attention recently because they have shown far superior results in recognition challenges and benchmarks that far exceed methods which are not based on neural network. AlexNet [11] is perhaps among the most prominent early examples, achieving a significant boost in the large scale visual recognition challenge [30]. In earlier years deep neural networks could not achieve such a performance because of a number of reasons. Perhaps the most important reason is that neural networks have a large number of parameters and therefore require very large data sets to prevent overfitting, such data sets were not available. Second, the advances in GPGPU programming and the availability of powerful GPUs has enabled the training of large deep learning models.

We note finally that are many variant of deep neural networks. When handling grid-like data such as images usually convolutional neural networks are used, recurrent

neural networks are used to handle sequences such as text [29]. Convolutional neural networks are a special case of deep neural networks. Using a deep network requires that every node be connected to all of the nodes in the previous layer, which leads to a very large number of weights which is more likely to lead to overfitting. In convolutional neural networks each node has local connectivity to the previous layer, leading to a much smaller number of weights. Further, because of translation invariance the weights are shared by a collection of units. This is a way to introduce prior knowledge in the model that leads to better performance for visual recognition tasks.

3.4 Review of Object Detection Using Deep Convolutional Neural Networks

The impressive results of AlexNet [11] which were shown on the image net large scale visual recognition challenge [30] has instigated a lot of research into deep learning based approaches in other visual recognition tasks. It was soon shown that significant improvements can be achieved in image segmentation [12] and object detection [7].

Although RCNN [7] led to a significant improvement in object detection, it has significant drawback in terms of the run times. Roughly, RCNN works by generating a set of proposals by a method such as selective search [22], the region of the image is warped to a fixed size, which is then forward propagated in a convolutional neural network, fc7 features are then pooled and used as features for an SVM which then does the classification and bounding box regression.

SPP-net [36] then showed that the whole image can be convolved and spatial pyramid pooling can be applied to collect features which are then fed to SVMs. Fast RCNN [6] has then shown that the classification and box regression can be done without the use of an SVM. Although this leads to a simpler architecture it does not solve the run time problem. Faster RCNN [16] showed a real time object detector by generating proposals using the introduced region proposal networks (RPNs) instead of the slow proposal generation methods such as [22]. Region proposal networks have an intriguing training procedure that leads to a collection of weights in the convolutional layer, each specializing in predicting proposals of a specific scale. As the small RPN network slides over the feature map, roughly positive labels are given if the intersection over union (IoU) between the ground truth box and the associated fixed scale box "anchor" is above a threshold and is negative if it is below a certain threshold. Since high IoU requires a significant match in the scale of the ground truth box and the anchor, thus positive labels will not be given if the scale mismatch is high. At test time each filter will predict proposals with scales around the scale of its associated anchor.

Even though the run time was significantly improved by Faster RCNN, there is still a heavy computational expense associated with pooling the features from every proposal region and then forward propagating them through two fully connected layers. Therefore, R-FCN [4] was introduced to significantly reduce that computational expense. In R-FCN a new convolutional layer is introduced consisting of $k^2(C+1)$ many feature maps, where C is the number of classes and k refers to the dimension of spatial grid. Each class (including the background) has k^2 many

position-sensitive feature maps associated with different positions in the image (top-left, top-middle, …, bottom-right). After the proposals are obtain instead of pooling then forward propagating through two fully connected layers, position-sensitive pooling followed by score averaging is done. A $(C+1)$-d vector is generated and used to predict softmax classification scores across the different classes. A similar layer and procedure is also introduced for regression.

3.5 Proposed Architecture

Since thumbnail generation is done by selecting a crop (bounding box) in the original image and then scaling it down to the thumbnail size. It is clear that the problem has a lot of connection with object detection. In fact, we model the problem as a bounding box prediction problem with two classes: representative of the image vs non-representative of the image. It should be noted that better results can be reached if the architecture is fully convolutional, which is the case in some object detection architectures such as R-FCN [4]. Using an architectures that it is not fully convolutional would require the input image to have a fixed size, if the image has an aspect ratio that it different from the fixed size aspect ratio, the downscaled image would need to be cropped. Cropping the input image is likely to produce below optimal results, since important regions in the image could have been cropped out.

Object detectors receive a single input, the image. However, in the case of automatic thumbnail generation, there are two inputs: the image and the given thumbnail size. In a line similar to that done by RPN and R-FCN which introduce specialized filters, we introduce a collection of filters that specialize in predicting crops for different aspect

ratios in the R-FCN architecture. A set of A points are introduced in the aspect ratio range from 0.5 to 2. The set represents aspect ratios that grow by a constant factor (i.e. a geometric sequence). The set $S = \{\frac{1}{2}c, \frac{1}{2}c^2, \ldots, \frac{1}{2}c^A\}$, with $\frac{1}{2}c^0 = \frac{1}{2}$ and $\frac{1}{2}c^{A+1} = 2$, leading to $c = \sqrt[A+1]{4}$. The filter banks in the last convolutional layer are modified into a set of A pair, with each pair having a total of $2k^2$ filters.

During training, when an image-thumbnail size pair is received, the image is forward propagated through convolutional layers up to the last convolutional layer. Based on the input thumbnail's aspect ratio an element in the set S is chosen (the one with the closest value). The loss is calculated for the pair associated with that element and set to zero for the others. The intersection over union (IoU) between the ground truth and the proposals is used to assign positive and negative proposal labels. Namely, the proposal label is negative unless the IoU ≥ 0.5. Similar to the classification branch, a regression branch is also utilized with A aspect ratio-specific regressors, with each regressor corresponding to an element in S. For a given proposal, the loss is calculated according to the given equation:

$$L(s_i, t_i) = \sum_{i=1}^{A} l_i L_{cls}(s_i, s^*) + \lambda[s^* = 1 \wedge l_i] L_{reg}(t_i, t^*)$$

where l_i is a binary variable with value 0 meaning ignore, and value 1 meaning factor-in, l_i values are assigned according to the following:

l_i ={ 1 $if\ argmin_i \left|\frac{1}{2}c^i - thumbail\ aspect\ ratio\right|$, 0 $otherwise$}

s_i is the ith pair's prediction of representativeness, s^* is the ground truth label, and L_{cls} is cross entropy loss. λ is a weight for the regression loss, which is set to 1. The classification and regression losses are zero except for the assigned element with $l_i =$

1. The smooth L_1 loss [6], is used for the regression loss, t_i is the ith regressor's bounding box prediction and t^* is the ground truth bounding box. The predictions are parametrized as done in [6]. The architecture of our model, which we call (Fast Automatic Thumbnail Generation) Fast-AT is illustrated in figure 7. Each aspect ratio regressor is responsible for a range of aspect ratios not a fixed value, therefore its predicted bounding box could have an aspect ratio that differs significantly from the thumbnail's aspect ratio. To avoid deformation when scaling the crop down to the thumbnail's size, the crop is rectified. The rectification step is simple, a bounding box with an aspect ratio equal to the thumbnail's is placed in the middle of the crop and scaled up until it touches the boundary of the crop. Because the difference in aspect ratio between the predicted box and the thumbnail's is not large, the change in the predicted box is not significant. This is shown in figure 8(b).

Resnet-101[8] is the backbone of our network, a learning rate of 0.001, momentum of 0.9, weight decay of 0.0005 with approximate joint training is used [16].

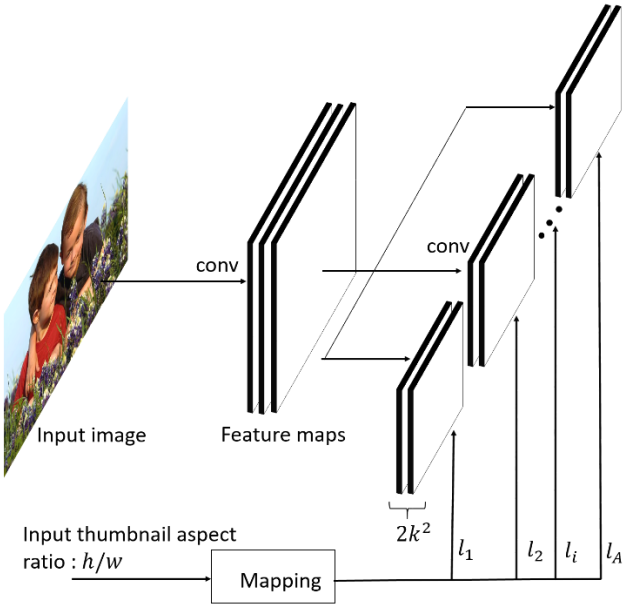

Figure 7. Diagram of the proposed architecture (Fast-AT). The filter is selected based on the input thumbnail's aspect ratio through the mapping function.

3.6 *Experiments*

The following metrics are used to compare the models:

• offset: the distance between the centers of the ground truth and predicted boxes. [9]

• rescaling factor (rescaling): defined as the max value of ratio of the rescaling factors between the thumbnail and ground truth, i.e. $\max(\boldsymbol{s_g}/\boldsymbol{s_p}, \boldsymbol{s_p}/\boldsymbol{s_g})$ where $\boldsymbol{s_g}$ and $\boldsymbol{s_p}$ are the rescaling factors for the ground truth and predicted box, respectively. [9]

• IoU: the intersection over union between the ground truth box and the predicted box.

• aspect ratio mismatch (mismatch): the square of the difference between the aspect ratios of the ground truth and the predicted box.

The data set consisting of 70,048 annotations over 28,064 is split into training and testing sets: 24,154 images with 63043 annotations for training and 3,910 images with 7,005 annotations for testing. We note that this is a 90% to 10% split between training and testing, respectively and that the two sets do not share any images.

Table 1 compares results between different trained models. We begin with R-FCN without modification. The architecture does not take the input thumbnail size into account, the number of classes is simply reduced to two, and further modifications are made to the architecture according to this reduction in class number. It is seen that R-FCN performs well, its metric results are good except for the mismatch metric where it performs badly. The high value in the mismatch metric indicates that the aspect ratios of the predicted boxes deviate significantly from the aspect ratios of the ground truth boxes, because of the large deviation values, the final thumbnails obtained by the rectification step then scaling down are likely to be different from the predicted boxes, causing the final thumbnail to miss important regions in the image. This is illustrated in figure 8(a).

Our proposed model is then considered, the aspect ratio is divided into 5 division, i.e. $A = 5$. Our model improves the overall metrics, with 4% improvement in IoU, and significant drop in offset and rescaling. The aspect ratio mismatch also drops by an order of magnitude when compared to the R-FCN model.

Another model we consider, extends the divisions into thumbnail sizes as well as aspect ratio leading to a total of 15 divisions, 5 aspect ratio divisions per thumbnail size. Small thumbnails (32-64) have 5 aspect ratio divisions and so do medium thumbnails (64-100), and large thumbnails (100-200). This model however, does not lead to an improvement over the model where only aspect ratio divisions are used.

Model	offset	rescaling	IoU	mismatch
R-FCN	56.2	1.192	0.64	0.102
Fast-AT (AR)	55.0	1.149	0.68	0.010
Fast-AT (AR+TS)	55.4	1.154	0.68	0.012

Table 1. Comparison between different models in terms of the metrics. R-FCN, Fast-AT with aspect ratio mapping (AR), Fast-AT with aspect ratio and thumbnail size mapping (AR+TS).

Figure 8. R-FCN and Fast-AT predictions in blue and the rectified box in red. With the final crop shown below (a): For R-FCN, it is clear that important regions in the image are missing in the final crop. (b): For Fast-AT predictions, the rectified box does not differ significantly from the original predicted box and the final crop still maintains the important regions in the image.

3.7 Comparison to other methods

Our method is compared to other thumbnail generation methods through metric and visual evaluations. In addition, a user study is conducted. The following methods are compared to ours:

• Scale and Object-Aware Saliency (SOAT): This methods computes a scale and object saliency map, then a greedy region search algorithm is used to find the crop [21].

• Efficient Cropping: This method computes a saliency map and conducts region search in linear time to find the crop. Unlike SOAT, the algorithm searches for crops with the specified aspect ratio. We note that when the aspect ratio mismatch between the image and the thumbnail is large, a solution may not exist (i.e., the problem is infeasible). The method is used without aspect ratio restriction is such a case. The saliency threshold value is set to 0.7, when running this method [2].

• Aesthetic Cropping: This method considers a candidate set of crops and picks the one with the largest aesthetic score [23].

• Visual Representativeness and Foreground Recognizability (VRFR): The objective of this method is similar to ours. However, it cannot generate thumbnails of arbitrary size, rather only thumbnails of size 160× 120 [9].

We note that the aesthetic method and VRFR did not release their code, so our comparison to them is limited to a user study, namely the 200 images with their generated thumbnails that were publically released by [9] are used.

3.7.1.Metric Comparisons to other models:

The different methods are compared using the same metrics that were used in the experiments section with the addition of two more metrics, the hit ratio h_r and the background ratio b_r [9], defined as:

$$h_r = \frac{|g \cap p|}{|g|}$$

$$b_r = \frac{|p| - |g \cap p|}{|g|}$$

where p is the predicted box and g is the ground truth box. The test set of 3,910 images with 7,005 annotations is used to evaluate the metrics (Table 2. Shows the performance of each method). The data set that we use is more challenging than the MIRFLICKR-25000 dataset [10] that was used in [9] having larger sizes with variation in quality and with many instances that include multiple objects. This leads to much higher offset values than those reported in [9].

Our method achieves the best performance in all of the metrics. We observe that the efficient cropping has a non-zero mismatch values, which is explained by the examples in the test set where the problem was infeasible according to the selected threshold value and the imposed aspect ratio restriction. SOAT has the highest aspect ratio mismatch values which is expected since it is agnostic to the input aspect ratio value.

The hit ratio indicates the amount the bounding box captures from the ground truth and the background ratio indicates the amount of the bounding box that lies outside the ground truth. The optimal method should predict a box which is very close to the

ground truth, and thus should have a large hit ratio and a small background ratio. In terms of hit and background ratios, the methods exhibit the same behavior as reported in [9]. Specifically, saliency based methods (SOAT and efficient cropping) predict crops that focus on small regions that have large saliency. This leads to low hit and background ratio values. In comparison, Fast-AT has a large hit ratio and a low background ratio. This indicates that the predicted boxes closely match the ground truth boxes.

Method	Offset	rescaling	IoU	Mismatch	h_r	b_r
SOAT	*80.5*	*1.378*	*0.52*	*0.204*	*68.7%*	*41.6%*
Efficient Cropping	*88.3*	*1.329*	*0.52*	*0.176*	*64.4%*	*34.3%*
Fast-AT	*55.0*	*1.148*	*0.68*	*0.010*	*83.7%*	*37.1%*

Table 2. Comparison of different automatic thumbnail generation methods. We only compare against saliency based methods here.

3.7.2. User Study and Visual Results:

We conduct a user preference study where we show the original image along with the generated thumbnails from different methods. The methods that are used are SOAT, Efficient Cropping, and Fast-AT. Users are asked to select the best thumbnail. A total of 372 images were picked randomly from the test set of 7,005 images. 30 users from AMT participated and every user was restricted to a total of 30 votes. The results of this study are shown in table 3. It is clear that Fast-AT has a much better performance in comparison to the other two methods.

SOAT	Efficient Cropping	Fast-AT
88 (23.7%)	86 (23.1%)	198 (53.2%)

Table 3. Number votes received by each method.

Another user study was performed, this time using the 200 images that were released by [9]. The comparison was between SOAT, aesthetic based cropping [23], VRVF [9], and Fast-AT. Table 4 shows the results of this study. Fast-AT and VRFR achieves a comparable performance with Fast-AT performing slightly better. We note that VRFR only works for thumbnails of size 160 ×120 and that it requires 60 second per thumbnail.

SOAT	Aesthetics Based Method	VRVF	Fast-AT
34(8.5%)	92(23%)	135(33.7%)	139(34.7%)

Table 4. Number of votes received by each method.

We further compare Fast-AT to SOAT and efficient cropping visually as shown in figure 9. Although saliency based methods succeed in capturing the important regions in the image, in some examples they can produce thumbnails with clear deformation. This is exhibited by SOAT in many examples and in some example of efficient cropping. We note also that saliency based methods may ignore the semantics of the scene and hence ignore important regions in the image. This is seen in the third and fourth SOAT thumbnails and in the first and second efficient cropping thumbnails. In comparison, Fast-AT succeeds in each example, producing thumbnails that tightly enclose the representative region of the image.

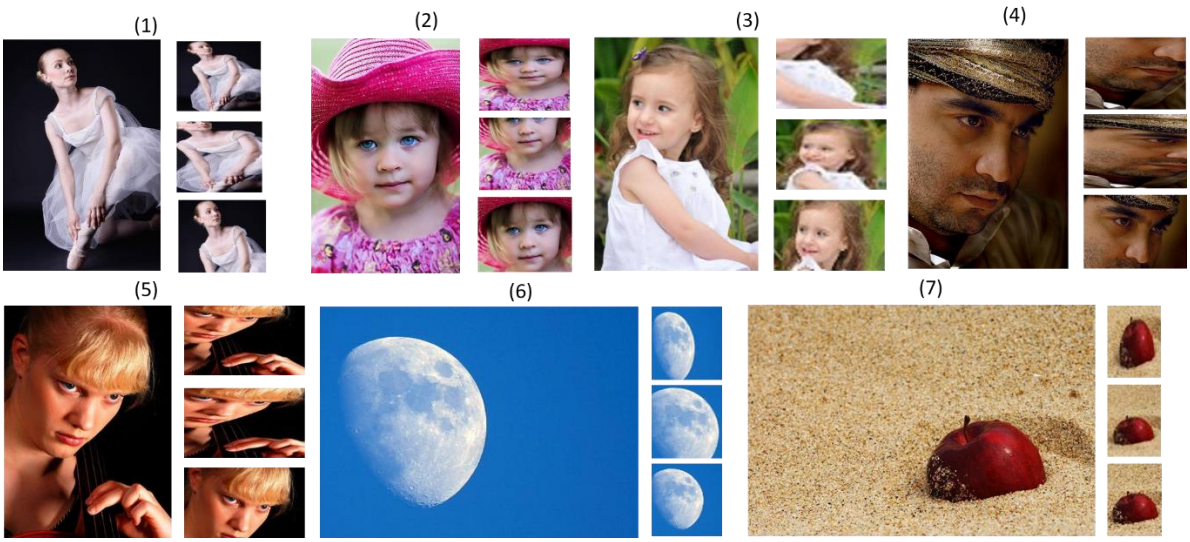

Figure 9. Examples of images and their generated thumbnails: The original image is on the left, to its right we display the thumbnails: top is SOAT, middle is efficient cropping, and bottom is Fast-AT.

3.8 Failure Cases and Multiple Predictions

To gain further insight into our model, we study the failure cases. In the test set, we look for examples where the IoU is below 0.1. Figure 10(a) shows some examples. It can be seen that although the prediction could be far from the ground truth, it can still capture representative regions in the image.

We also look at predations with the second or third highest confidence. It can be seen in figure 10(b) that these predictions can be close to the ground truth. Therefore, if the system is to be deployed, users can find it useful if the system outputs a small set of predictions instead of just one. Users can pick the best thumbnail from the set of candidate thumbnails. We further test the performance of the model when the second and third most confident predictions are used. We find that the second predictions

leads to a significant improvement, but that is not the case in for the third prediction.

The results are shown in table 5.

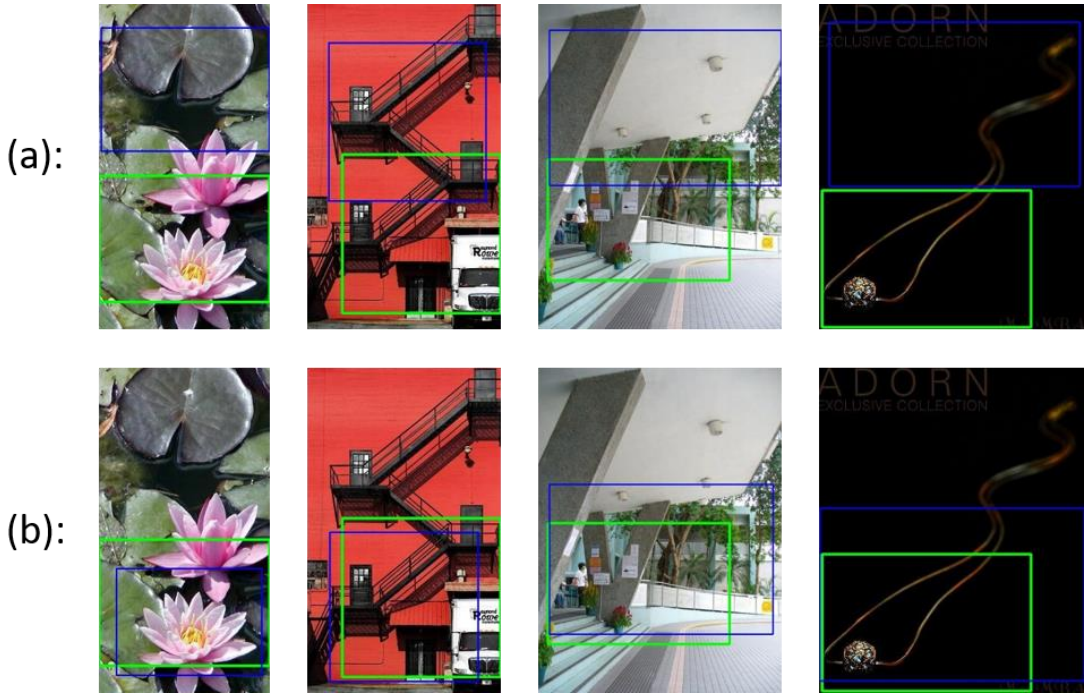

Figure 10. Failure cases of Fast-AT, the ground truth is in green and the prediction is in blue:(a): although the prediction could have low IoU with the ground truth but still capture a representative region. (b): the second or third most confident prediction can be close to the ground truth.

Model	offset	rescaling	IoU	mismatch
Top 1	55.0	1.149	0.677	0.010
Top 2	50.4	1.152	0.693	0.011
Top 3	50.3	1.152	0.693	0.011

Table 5. Fast-AT performance when using top 1, 2, and 3 predictions. Using the top 2 predictions leads to a significant improvement in the offset and IoU. Using the top 3 predictions does not lead to a significant improvement.

Multiple predictions can also be useful when the aspect ratio mismatch between the thumbnail and the representative part of the image is significant. The crop's aspect ratio is constrained to the thumbnail's aspect ratio and therefore it may not be able to capture all of the representative region in the image. Multiple prediction made by Fast-AT can cover different representative regions in the image. This is shown in figure 11. The region of interest in the first three images (from the left) is spread horizontally (best captured by a wide thumbnail), but the input thumbnail is tall. The reverse is true for the last image. In the first row the prediction with the highest confidence is shown and in the second row the prediction with the second highest confidence is shown. It is clear, that using multiple predictions is useful in capturing different representative regions in the image.

Figure 11. There is a significant aspect ratio mismatch between the representative region of the image and the input thumbnail's aspect ratio. In the first row, the most confident prediction is shown, while the second prediction is shown in the second row. It is clear that they are useful in capturing different representative regions in the image.

Chapter 4: Conclusion and Future Work

In this paper we have improved over the existing baselines for automatic thumbnail generation. Unlike previous solutions, our proposed solution does not depend on saliency or heuristic considerations but rather addresses the problem directly. We collected a data set consisting of 70,048 thumbnail annotations over 28,064 images. We trained a CNN which makes predictions in real time using this set. Our solution has shown superior performance in comparison to other methods as demonstrated by the metric evaluations as well as a user study. We have further investigated with failure examples of our model and have seen that in some failure cases the second and third predictions can be close to the ground truth.

Each image has an optimum thumbnail aspect ratio which depends on how the representative region of the image is spread, this can be seen in figure 11. In the in the first 3 images, using a tall aspect ratio thumbnail would require a tall crop which cannot cover the whole representative region of the image which is wide. Our model currently, cannot be used to predict the ideal aspect ratio. To accomplish this, a data set of images and their ideal thumbnails has to be collected. In our original data set the selected crops were forced to have an aspect ratio that it is equal to the thumbnail's aspect ratio. The bounding box should not have a restricted aspect ratio if the ideal aspect ratio is to be obtained.

Another interesting problem is how to optimally display a collection of thumbnail images. In this problem we were focused on a single image, not multiple images. Specifically, given a fixed display size and a collection of images to be displayed in a

non-specified layout, what is the optimum layout and where does each thumbnail image fit?

Finally, it seems possible to extend the model for aesthetic based cropping. However, there is no large data set for this problem which is required for a deep model. Moreover, such data sets tend to be expensive to collect since it would need to be annotated by experts [23].

Bibliography

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2189–2202, 2012.

[2] J. Chen, G. Bai, S. Liang, and Z. Li. Automatic image cropping: A computational complexity study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 507–515, 2016.

[3] G. Ciocca, C. Cusano, F. Gasparini, and R. Schettini. Selfadaptive image cropping for small displays. IEEE Transactions on Consumer Electronics, 53(4):1622–1627, 2007.

[4] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409, 2016.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. IEEE, 2009.

[6] R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.

[9] J. Huang, H. Chen, B. Wang, and S. Lin. Automatic thumbnail generation based on visual representativeness and foreground recognizability. In Proceedings of the IEEE International Conference on Computer Vision, pages 253–261, 2015.

[10] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval, pages 39–43. ACM, 2008.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015.

[13] W. Luo, X. Wang, and X. Tang. Content-based photo quality assessment. In 2011 International Conference on Computer Vision, pages 2206–2213. IEEE, 2011.

[14] J. Mannos and D. Sakrison. The effects of a visual fidelity criterion of the encoding of images. IEEE Transactions on information theory, 20(4):525–536, 1974.

[15] M. Nishiyama, T. Okabe, Y. Sato, and I. Sato. Sensation based photo cropping. In Proceedings of the 17th ACM international conference on Multimedia, pages 669–672. ACM, 2009.

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[17] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. In ACM transactions on graphics (TOG), volume 29, page 160. ACM, 2010.

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

[19] F. Stentiford. Attention based auto image cropping. In Workshop on Computational Attention and Applications, ICVS, volume 1. Citeseer, 2007.

[20] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs. Automatic thumbnail cropping and its effectiveness. In Proceedings of the 16th annual ACM symposium on User interface software and technology, pages 95–104. ACM, 2003.

[21] J. Sun and H. Ling. Scale and object aware image thumbnailing. International journal of computer vision, 104(2):135– 153, 2013.

[22] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.

[23] J. Yan, S. Lin, S. Bing Kang, and X. Tang. Learning the change for automatic image cropping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 971–978, 2013.

[24] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision, pages 818–833. Springer, 2014.

[25] Avidan, Shai, and Ariel Shamir. "Seam carving for content-aware image resizing." *ACM Transactions on graphics (TOG)*. Vol. 26. No. 3. ACM, 2007.

[26] Rubinstein, Michael, Ariel Shamir, and Shai Avidan. "Multi-operator media retargeting." *ACM Transactions on Graphics (TOG)*. Vol. 28. No. 3. ACM, 2009.

[27] Wang, Yu-Shuen, et al. "Optimized scale-and-stretch for image resizing." *ACM Transactions on Graphics (TOG)*. Vol. 27. No. 5. ACM, 2008.

[28] Santella, Anthony, et al. "Gaze-based interaction for semi-automatic photo cropping." *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006.

[29] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[30] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision* 115.3 (2015): 211-252.

[31] http://neuralnetworksanddeeplearning.com/chap5.html .

[32] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[33] Kawaguchi, Kenji. "Deep learning without poor local minima." *Advances In Neural Information Processing Systems*. 2016.

[34] Telgarsky, Matus. "Representation benefits of deep feedforward networks." *arXiv preprint arXiv:1509.08101* (2015).

[35] He, Kaiming, et al. "Identity mappings in deep residual networks." *European Conference on Computer Vision*. Springer International Publishing, 2016.

[36] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015): 1904-1916.

[37] Esmaeili, Seyed A., Bharat Singh, and Larry S. Davis. "Fast-AT: Fast Automatic Thumbnail Generation using Deep Neural Networks." *arXiv preprint arXiv:1612.04811* (2016).