You've got nothing on me! Privacy Friendly Face Recognition Reloaded

Stephan Escher, Patrick Teufert, Lukas Hain, and Thorsten Strufe

TU Dresden, Germany
{firstname.lastname}@tu-dresden.de

Abstract. Nowadays, almost anyone can take pictures at any time. Simultaneously, services such as social networks make it easy to share and redistribute these images. Users who do not want pictures of them to be recorded and distributed can hardly defend themselves against this. With the introduction of the GDPR in the European Union, users can now at least demand the deletion of such unsolicited uploaded data from web platforms. To find such images, however, the user must first upload comparative images to such a web service so that this service can compare them with its database to show the user whether unwanted images exist or not. This means that the user must involuntarily pass on his biometric data to a web service where he does not actually want his data to be saved. Thus, in this paper, we present our privacy-friendly face recognition approach based on Local Binary Patterns and Error Correction Codes, that allows users to query web services for the presence of unwanted images without revealing biometric information. We evaluated each step of our approach with the "FERET database of facial images" and the "Yale Face Database".

Keywords: privacy · face recognition · biometric data

1 Introduction

Nowadays, it is possible for anyone to take high-quality snapshots at any time thanks to a wide range of high-quality and affordable image capture devices. In addition, various Internet services enable the sharing, publishing and storage of these recordings for a global audience. However, the constant presence of visual recording devices such as smartphones, compact cameras or AR-glasses, also affects the privacy of users, who have no possibility to prevent unwanted capturing. Further, the unwanted upload of video and image material to platforms such as YouTube or Facebook can not be controlled by the user.
On Friday 25$^{\text{th}}$ May, 2018 the GDPR became active in the European Union [1], which enables users to request the deletion of their digital data on web platforms. However, even though the user now has the right to have these unauthorized uploaded data deleted, he must first find it.
In this paper we consider a scenario where a user wants to know if a particular web service contains images that depict him. To do this, currently, he must either

browse this service manually or he has to upload comparative images of himself so that the service can process and compare them with images in its database. The latter means that in order to find and delete unwanted images, the user must first of all offer his own images to this service. Overall, this process could result in restrictions regarding privacy, esp. if the service does not have any data of the user.
An example could be the "Non-Consensual Intimate Image Pilot" by Facebook. This is a program that prevents individuals from uploading intimate pictures of others to the Facebook platform without their consent. This requires individuals "to establish which image is of concern by sending the image to themselves on Messenger". Then a "specially trained representative reviews and hashes the image". The hash is saved and the photo is finally deleted from the messenger and Facebook's servers [2]. It's easy to imagine that individuals might have concerns about uploading intimate photos of themselves on Facebook in the first place.
In general, users may not want to share private pictures with a platform provider they do not trust. Thus, in this paper, we propose our approach for privacy-friendly comparisons of facial images and thus for privacy-friendly requests to web services without leaking biometric data. We focus on an approach that is easy to implement, so that even smaller medium-sized companies can offer this solution.

2 Related Work

Erkin et al. [6] tried to solve this scenario with "Homomorphic Encryption" and the use of "Eigenfaces" for face recognition. They achieve good results in facial recognition and the protection of privacy, but the use of "Homomorphic Encryption" is computationally intensive and also has a high communication complexity between server and user. A desirable feature of "Homomorphic Encryption" is that neither the user needs to share private images nor does the server provider need to share its database or recognition model. A comparable approach is that of Sadeghi et al. [13], where they also use "Homomorphic Encryption" and build on that to achieve faster computation and less complexity.
Another approach was attempted by Chanyaswad et al. [5] using "Eigenfaces" to retrieve vectors representing facial images and Discriminant Component Analysis (DCA) to reduce the dimensions of such a vector in order to protect the privacy of users. Chanyaswad et al. analyzed a problem scenario where training data (image vectors) are uploaded into the cloud by the user to classify the training data, while the cloud provider is malicious and tries to reconstruct the original images from the training vectors received from the users. They achieved facial recognition results above 90% recognition rate and also that the images cannot be recovered by the provider. One disadvantage of this approach is that a training phase is required, which can be difficult for providers of small image databases as well as for users.

3 Towards Privacy Friendly Face Recognition

Our approach is based on a traditional Face Recognition algorithm using local binary pattern histograms (LBPH). The advantage of the usage of LBPH lies in its easy calculability as well as in its good ability to differentiate. In addition, a feature representation can be generated directly from an image and compared to others, without the need of training data or a training process. After quantizing the LBPH results to a binary representation we use the fuzzy commitment scheme for privacy friendly distance calculation. Furthermore we consider the possibility to apply error correction directly to the quantized LBPH representation to get a unique value for each person, instead of comparing the distance between two representations. The pipeline of our approach could be seen in Figure 1.

Fig. 1. Pipeline of the Privacy Preserving Face Recognition Approach

We assume a face detection algorithm, which can detect embedded faces in images of the user and the service [14]. By applying the pipeline to detected faces, facial features of the user can be compared privacy-friendly with all available ones of the web service.

3.1 Preprocessing

First of all the images are *aligned*, based on the eye coordinates given in the data sets. Therefore the image is rotated clockwise until the left and right eyes are on the same horizontal line parallel to the x-axis. Then the image is *cropped* so that the image consists only of the person's face. This is done in relation to the eye-center, calculated by the given eye coordinates. Next the aligned and cropped image is enhanced via *histogram equalization*, which was mentioned by [3]. Finally the images are resized to a fix size (depending on the LBPH settings). The whole preprocessing pipline can be seen in Figure 2.

3.2 Local Binary Pattern Histograms

LBP is a type of visual discriptor, which describes the texture and local characteristics of an image. Therefore the original image is first divided into equal-sized blocks. A histogram is generated per block, which describes the relationship of the pixels to their neighbours. For this the grey value of each pixel in a block is compared with its P neighbours which lie in the radius R. Usually $P = 8$ and $R = 1$ which represents the eight direct neighbours of a pixel. If the grey value is higher then the neighbour value it returns 1 otherwise 0. This is done

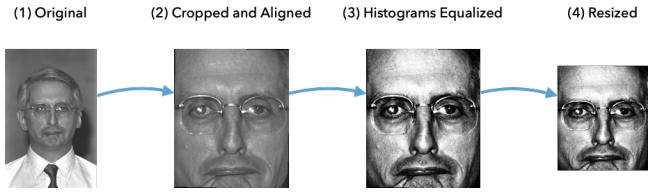

Fig. 2. Preprocessing Steps

clockwise for each P neighbours, resulting in a binary pattern of length P for each pixel (see Figure 3). The frequency of all patterns obtained are transferred in a histogram with 2^P bins. Finally the LBPH feature vector for the image is recieved by concatenating the histograms of each block.
One way to reduce the number of bins per histogram is to use uniform LBP, an extension of LBP. There the histogram only includes binary patterns where all 1 and all 0 are adjacent. These patterns can be interpreted as corners, edges or areas. All other combinations are summarized as 'non-uniform'.
In our setting we used two different LBPH approaches for face recognition. The first one is the original approach from Ahonen et al. [3]. They resized the original images to (130, 150) pixels. Afterwards they divided each image into blocks of size (11,13) and get the best results with the extraction of uniform patterns with a neighbourhood of eight and a radius of two ($P = 8, R = 2$).
The second one is an improvement of the LBPH face recognition approach by Girish et al. [10]. They use multi-block LBP, which is similar to LBP but instead of individual pixel values, summed blocks are used to generate the patterns. They resized each image to a size of (92, 112) and divided it into blocks of the size (23, 28). See also Figure 4 right.

Fig. 3. Left: example binary pattern creation. Right: example impact of P and R value.

3.3 Quantization

To get a binary representation from the histogram feature vector we will quantize it. In general to quantize, we create a threshold and check whether a histogram bin is greater than the threshold. If so, we rewrite this histogram bin as 1,

otherwise as 0. This way we generate a binary representation of the histogram values. For the threshold we use the mean of the histograms, the global and local (for each histogram of a block) median as well as the relation between histogram values (see Figure 4 left). Further we used a double bit quantization (DBQ) [9].

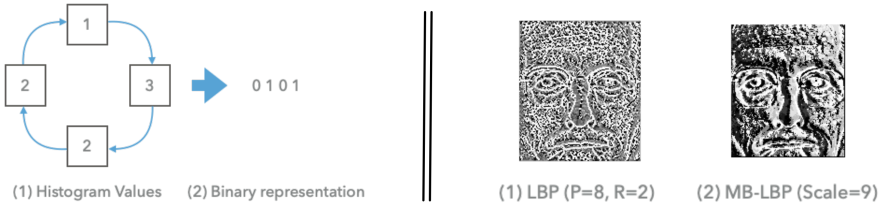

Fig. 4. Left: example of relation quantization. Right: example of LBP representations (1) uniform LBP ($P = 8, R = 2$) (2) MB-LBP (Scale = 9)

3.4 Usage of Error Correction Codes

The first approach we took into consideration was the direct use of an Error Correction Code (ECC) on the binary representation of a facial image, so that each face is mapped to a unique value [8].
An error correction code described by the parameters (n, l, d_{min}) encodes information words of length l and adds $k = n - l$ redundant bits to the code words of length n. Binary codes can detect $f_e = d_{min} - 1$ errors in a distorted word $b = a \oplus e$ where e is the error word and a the information word. A distorted word can only be reconstructed if the error word e has a weight of $f_k = \lfloor (d_{min} - 1)/2 \rfloor$. Assuming that the quantized LBPH representation of a facial image x is a distorted code word, we propose the utilization of a decoding $dec()$ function to correct the errors and to map the representation to a unique value. The f_k parameter of an error correction code is defined according to d_{min} and the error correction code aims to map a noisy channel code word to a *valid* one. However, in our setting we aim to map two noisy channel code words to a single valid one. We are able to map both noisy channel code words to the same valid channel code word as long as the sum of the hamming distance between a noisy channel code word and a valid channel code word and the hamming distance between the two noisy channel code words is smaller than f_k . Furthermore, if the number of bit errors is greater than f_k for each true negative, the decoded values with length $\lfloor \frac{length(x)}{n} \rfloor \cdot l$ might be used to identify a person in a set of images. If we find a matching ECC we can use this setting for a privacy-friendly Face Recognition approach.
In our setting, the client calculates the LBPH pattern of their image to compare, quantizes it (x), corrects the errors and decodes the binary representation with the decoding function of the error correction code into the unique information

word ($dec(x)$). This value is sent to the server after hashing it with a cryptographic hash function $y = h(dec(x))$. The server processes all facial images in its database in the same manner, computing $y' = h(dec(x'))$. Upon receiving y, the server determines whether or not y is part of its database ($y \stackrel{?}{=} y'$) and consequently presents all corresponding images.

3.5 Fuzzy Commitment

A second approach is the usage of a fuzzy commitment scheme by Juels and Wattenberg [7]. For the usage for a privacy-friendly face recognition approach we use the fuzzy commitment described as follows.
The user chooses a random code word c and calculates the quantized LBPH representation of an facial image x. After that he calculates $d = c \oplus x$ and computes a hash of the random code word c with a cryptographic hash function $h(c)$. To figure out if unwanted images of the user are uploaded to a specific web service, he sends $(h(c), d)$ to the aforementioned server. Note that this value does not include facial information about the user.
The server calculates for all his images x': $y = d \oplus x' = x \oplus x' \oplus c$. If x' is the binary LBPH representation of the same person as x, y only contains the errors caused by the LBPH algorithm and c. After that a decoding function of an Error Correction Code $dec()$ could correct these errors which should map back to c: $c' = dec(y) = dec(x \oplus x' \oplus c)$. For all $h(c) = h(c')$ the server is able to present the corresponding images, because they contain the same person the user was looking for.

4 Experimental Analysis

4.1 Experiment Setup

For the evaluation part we have used the Feret[1] [11, 12] and the Yale data set [4]. For Feret we selected all images of a person from the "fa" and "fb" part of the data set. We created pairs for each person between each image of that person in "fa" and the images in "fb", in a way so that each image is only compared ones. In addition, we randomly selected images of different people and created a pair of images from their pictures.[2]
We used a different test setting for the Yale data set, as it contains fewer people, but each person is photographed in more poses. We randomly selected a picture of each person and created pairs with every other picture of that person. In addition, the randomly selected image of one person is compared with every image of the other persons in the same pose.[3]

[1] "Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office"
[2] Resulting in an evenly distributed baseline of 50% true cases and 50% false cases.
[3] Resulting in a baseline of $\approx 61.11\%$ true cases and $\approx 38.89\%$ false cases.

The comparison for an image pair is performed using a distance metric. To evaluate the different steps, we use different distance metrics for the LBPH and binary representations of facial images. For LBP representations we use the "Histogram Intersection" [3] and for binary representations after quantization we use the "Hamming distance".
The calculated distances are used to decide whether a particular pair of images shows the same person or different people. This is done by evaluating whether the calculated distance is smaller than the threshold value. A threshold was found for each algorithm by performing various tests and selecting the threshold resulting in the best face recognition performance. In our scenario, a threshold is necessary because we are not trying to identify the person in one picture, but to find all pictures of a person.

4.2 Face Recognition with Local Binary Patterns

We used the configuration of the original LBPH approaches [3, 10] (see 3.2) as baseline LBP variants for our privacy friendly face recognition approach.
In a first step we analyzed this LBPH baseline regarding their recognition rate. The results are shown in Table 1. We found that the approach by [10] outperformed the one from Ahonen et al. [3] overall slightly. In general the results show that both approaches could well used to distinguish people through frontal facial images and to correctly match two face images if they depict the same person.

Table 1. Results of the original LBPH approaches, showing the average results for the Yale and Feret data sets.

Variant	Accuracy	Precision	F1-score	AUC
LBP[3]	0.8363	0.8235	0.8685	0.9318
MB-LBP[10]	0.8732	0.8667	0.8947	0.9462

4.3 The Impact of Quantization on Recognition Performance

We found that the application of quantization to the LBP representations has a higher negative impact on the MB-LBP variant [10] than on the original LBP approach. Table 2 shows the results for the local median quantization for both approaches. The approach of Ahonen et al. [3] is even more powerful when considering the "Accuracy", "Precision" and "F1-score", while only the "AUC value" decreases by less than one percent. This was unexpected, since by quantizing the LBP representation we essentially lose information about certain patterns and abstract the occurrence of patterns to a simple binary number.
A possible explanation would be that the original LBPH vectors contain unnecessary and redundant information which is reduced by the quantization and thus

improved the recognition rate. Otherwise the MB-LBP patterns already averaged the feature information in their approach, resulting in worse results after a further simplification through quantization.
Due to the original LBP variant being more robust to quantization we use this approach for further steps.

Table 2. Local median quantized LBP variants. The results represent the average results for the Yale and Feret data set.

Variant	Accuracy	Precision	Recall	F1-score	AUC
LBP[3]	**0.8759**	**0.9145**	**0.8629**	**0.8873**	**0.9298**
MB-LBP[10]	0.7787	0.8426	0.7490	0.7919	0.8522

4.4 Quantization Variants

The results for the different quantization methods used for the original LBP approach by [3] can be seen in Table 3.
The best results were obtained with the "Double-bit Quantization" [9], with the local median quantization taking second place for almost all metrics.
Although DBQ offers the best performance, it also uses twice the amount of binary values for a representation. The reason for this is that DBQ encodes each value with two bits during quantization, which, due to the much larger binary representation, results in higher computing power demands and problems when using "fuzzy commitment" [7] and other error correction codes. For this reason, and because of the only slight improvement over the "Local Histogram Median", we continue using the local median as quantization approach.

Table 3. Comparison of quantization approaches used for the configuration by [3]. The results represent the average results for the Yale and Feret data set.

Quantization	Accuracy	Precision	Recall	F1-score	AUC	Bit-Length
DBQ	**0.8883**	0.9215	**0.8793**	**0.8997**	**0.9340**	14278
Relation	0.8740	**0.9263**	0.8477	0.8853	0.9251	7139
Mean over Histogram	0.8522	0.8815	0.8575	0.8691	0.9158	7139
Global Histogram Median	0.8700	0.9141	0.8506	0.8808	0.9294	7139
Local Histogram Median	*0.8759*	*0.9145*	*0.8629*	*0.8873*	*0.9298*	7139

4.5 Usage of Error Correction

Before we can apply a decoding function of an ECC to the quantized LBPH representations we first analyse the error rate and shape of the representations. The

error rate is analysed regarding representations of facial images which contain the same person. Therefore, we aim to choose an ECC that meets the requirements such that different binary representations of the same person are decoded into the same information word.
The results (see Table 4) show that the differences between binary representations of the same person are quite high and the distribution of the binary values is not balanced, which is not a good starting point considering the application of an ECC. The higher error rate for the Yale data set is based on the fact that the facial images show more different poses of the persons than in Feret.

Table 4. Analysis of the binary representations with local median quantization.

Data Set	Avg. Hamming Dist.[4]	Std.[5]
Feret	0.1233	0.4898
Yale	0.1778	0.4747

We choose the $BCH(255, 9, 127)$ code as it has a low code rate of $\frac{l}{n} = 0.035$ and therefore a high error correction capability. The chosen BCH code maps 255 bits to a 9 bit information word and is capable of correcting $f_k = 63$ errors $(24,706\%)$. This setting allows at least for the Feret data set a twice as high error correction capability than the average error between two images of the same person. This might enable a mapping of two noisy channel code words to a single valid one. The binary representation of each image is split into $255bit$ blocks, remaining bits are discarded. After the aforementioned processing steps, the decoding function is applied to each block and the computed source code words are concatenated, resulting in a $243bit$ information word. The results of the ECC approach applied to the Yale Dataset are depicted in Table 5.

Table 5. Results with BCH(255, 9, 127) applied to local median quantized LBPH pattern with the configuration of [3].

TPR	TNR	FPR	FNR	Accuracy	Precision
0.1152	1.0000	0.0000	0.8848	0.4593	1.0000

The results show that using a decoding process with a BCH(255, 9, 127) and

[4] Average Hamming Distance means the average distance between all binary representations for the images of a person.
[5] The standard deviation shows the distribution of ones and zeros in the binary representation. A std. of 0.5 means that the distribution is equally divided.

our face recognition pipeline cannot recognize facial images in a desirable way. The accuracy is below the baseline and shows that almost every image pair is marked as not coming from the same person, as indicated by the false negative rate (FNR). On the positive side, there are also no false positive results (FPR), while there are still some true positive results (TPR). This means that some images of the same face in different poses have been decoded into the same information word.
By analysing the error rate for each block of a binary representation we can determine the reason for the result. Overall we recognize an average of 45 errors in each block between two noisy channel code words (binary representations) of the same person. On average, approximately 25% (7) of these blocks have more errors than can be corrected ($f_k = 63$). Additionally, not only the errors between two noisy representations must be corrected but also the errors between a noisy channel code word and the nearest valid one. As a result, the decoding process is not able to generate the same value for two images of the same person. However, in the best case, at the moment, we decoded three binary representations of the same person to the same value.
Better pre-processing, quantization or extended LBPH variants could improve the error rate of binary representations of the same person and with the choice of an appropriate ECC the approach could nevertheless be feasible, which should be evaluated in the future.

4.6 The Fuzzy Commitment Approach

To evaluate the "Fuzzy Commitment" approach we used the python implementation by Burkert[6]. Using this implementation we obtain a decision whether two binary LBPH representations are equal by subjecting an image of an image pair to the "Fuzzy Commitment" scheme, which results in a commitment. We then use this commitment and the comparative image to obtain a decision (see Listing 1.1).

Listing 1.1. Example usage of "Fuzzy Commitment"

```
tolerance = threshold * length
cs = FCS[list](length, tolerance, polynomial=32771)
commitment = cs.commit(original_binary_representation)
decision = cs.verify(commitment, comparative_binary_representation)
```

For the ECC we used a BCH Code using the polynomial 32771_{10}. The tolerance parameter describes the error correction capabilities of the BCH Code. In our case, we use the optimal threshold of the local median quantized LPB variant received via the hamming distance metric in previous tests to calculate the optimal tolerance of the ECC. The threshold indicates whether two images depict the same person or not.
The results are shown in Table 6. As it can be observed, with this setup, the use of "Fuzzy Commitment" achieves the same results as the quantized LBPH

[6] https://github.com/cburkert/fuzzy-commitment

approach.
This demonstrates the possibility of using "Fuzzy Commitment" for a privacy-friendly face recognition approach that compares binary representations of images.

Table 6. Fuzzy Commitment vs. LBP (local median Quantization, Hamming Distance)

Variant	Accuracy	Precision	Recall	F1-score
LBP [3]	0.8759	0.9145	0.8629	0.8873
"Fuzzy Commitment"	0.8759	0.9145	0.8629	0.8873

5 Conclusion

In this paper we presented an approach to allow users to query a web service for unwanted uploaded images without having to reveal their own image data. Thus we proposed a privacy friendly face recognition approach based on Local Binary Pattern Histograms (LBPH) and Error Correction Codes (ECC).
We have described our pre-processing pipeline, which was applied to two different data sets (Feret and Yale). After comparing two traditional LBPH face recognition approaches we quantized the results to obtain a binary representation of the facial features. We analyzed different quantization methods and showed that a local median quantization over histogram values shows the best results for our use case. Additionally we have shown that the use of quantization can preserve the results of LBP variants and does not diminish the results. In order to evaluate the possibility of a privacy friendly face recognition, we used the "Fuzzy Commitment Scheme" and showed that our quantized LBP representations can be used for "Fuzzy Commitment" without disturbing the face recognition results. This means that this approach can be used for a privacy friendly comparison of facial images and thus for a privacy friendly request to web services without leaking own biometric data. In addition, we propose to use ECC directly and analysed the use of a $BCH(255, 9, 127)$ code. We show that it is generally possible to decode image hashes from one person to the same hash. However, not yet practically applicable with our quantized LBPH features.
An important improvement in the future would be to reduce the average Hamming distance in the binary representations of quantized LBP representations, which would lead to easier use of error correction codes. In addition, other LBP variants may be able to improve the information displayed in a single image window by using more sophisticated algorithms, which could reduce the overall hash size. Quantization methods should be further analyzed to develop such quantizations that the histograms are binarized in a way that leads to good facial recognition results, a balanced binary distribution and a lower error rate between binary representations of two images of the same person.

References

1. EUR-Lex - 32016R0679 - EN, https://eur-lex.europa.eu/eli/reg/2016/679/oj, library Catalog: eur-lex.europa.eu
2. The Facts: Non-Consensual Intimate Image Pilot, https://about.fb.com/news/h/non-consensual-intimate-image-pilot-the-facts/, library Catalog: about.fb.com
3. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Recognition with Local Binary Patterns. In: Computer Vision - ECCV (2004)
4. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence (1997)
5. Chanyaswad, T., Chang, J.M., Mittal, P., Kung, S.Y.: Discriminant-component eigenfaces for privacy-preserving face recognition. In: 26th International Workshop on Machine Learning for Signal Processing (MLSP) (2016)
6. Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., Toft, T.: Privacy-Preserving Face Recognition. In: Privacy Enhancing Technologies (2009)
7. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Proceedings of the 6th conference on Computer and Communications Security - CCS (1999)
8. Kerschbaum, F., Beck, M., Schönfeld, D.: Inference Control for Privacy-Preserving Genome Matching. arXiv:1405.0205 (2014)
9. Kong, W., Li, W.J.: Double-Bit Quantization for Hashing. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
10. N, G.G., C.L., S.N., Das, P.K.: Face recognition using MB-LBP and PCA: A comparative study. In: International Conference on Computer Communication and Informatics (2014)
11. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.J.: The FERET database and evaluation procedure for face-recognition algorithms. Image and Vision Computing (1998)
12. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence (2000)
13. Sadeghi, A.R., Schneider, T., Wehrenberg, I.: Efficient Privacy-Preserving Face Recognition. In: International Conferenc on Information, Security and Cryptology – ICISC (2010)
14. Zafeiriou, S., Zhang, C., Zhang, Z.: A survey on face detection in the wild: past, present and future. Computer Vision and Image Understanding (2015)