IC-ChipNet: Deep Embedding Learning for Fine-grained Retrieval, Recognition, and Verification of Microelectronic Images

Md Alimoor Reza and David J. Crandall
Luddy School of Informatics, Computing, and Engineering
Indiana University, Bloomington, Indiana
{mdreza,djcran}@iu.edu

***Abstract*—Modern electronic devices consist of a wide range of integrated circuits (ICs) from various manufacturers. Ensuring that an electronic device functions correctly requires verifying that its ICs and other component parts are correct and legitimate. Towards this goal, we investigate using machine learning and computer vision to identify and verify integrated circuit packages using visual features alone. We propose a deep metric learning approach to learn a feature embedding to capture important visual features of the external packages of ICs. We explore several variations of Siamese networks for this task, and learn an embedding using a joint loss function. To evaluate our approach, we collected and manually annotated a large dataset of 6,387 IC images, and tested our embedding on three challenging tasks: (1) fine-grained retrieval, (2) fine-grained IC recognition, and (3) verification. We believe this to be among the first papers targeting the novel application of fine-grained IC visual recognition and retrieval, and hope it establishes baselines to advance research in this area.**

Fig. 1: We propose to learn deep embeddings from images of integrated circuits to retrieve, recognize, and verify them. Some instances from our dataset are shown.

I. Introduction

Recent reports in the popular press [1], [2] have highlighted the rising threat of hardware-based trojans in the devices that we rely on everyday. While much attention has been paid to software vulnerabilities, electronic devices that include counterfeit or defective integrated circuits could cause our digital devices, from cell phones to electric power stations to national defense systems, to fail prematurely or even to be subverted by malicious actors. Developing automatic systems that can help identify suspicious hardware components is thus critical for ensuring the safety and security of our electronic infrastructure.

One might assume that simply ordering integrated circuits (ICs) from a trusted supplier would ensure that the devices are legitimate. Unfortunately, ICs traverse a complex global supply chain between when they are manufactured and when they reach a customer [3]. For an example of what can go wrong, SparkFun [4], a popular and well-respected supplier of electronic parts for hobbyists, started receiving complaints about one of the microcontrollers they were selling. After much investigation [5]–[7], including inspecting the silicon of the IC with an electron microscope, they discovered that they had inadvertently bought a batch of counterfeit ICs: it appeared that someone had bought or found cheap, discarded ICs, removed the markings from the packaging, and then re-labeled and sold them as a much more expensive microcontroller. These fake parts could have been injected into the supply chain at numerous points: the highly distributed nature of the modern supply chain makes it efficient and resilient, but also makes it increasingly difficult to trace the provenance of any given part.

In lieu of somehow changing the supply chain, one viable option to reduce the threat of counterfeit parts is to develop automated systems that analyze the visual appearance of critical components such as integrated circuits (ICs) and verify that they appear as expected. Ideally, such algorithms would cue on subtle variations in the appearance of an IC that could indicate a counterfeit part: uneven silk screening, unexpected or missing markings, incorrect manufacturer logo, etc. [3] A challenge in doing this is that most of the recent success in computer vision, including image recognition [8], image detection [9], semantic segmentation [10], etc., considers relatively coarse-grained recognition problems in consumer images, whereas spotting counterfeit parts requires spotting subtle cues. Moreover, unlike consumer images and other popular applications, training data for counterfeit products is

978-1-7281-8243-8/20/$31.00 ©2020 IEEE

severely limited.

In this paper, we propose computer vision techniques for microelectronics security applications. In particular, we propose a deep neural network for learning an embedding from the images of Integrated Circuit (IC) chips commonly found in microelectronic devices. The learned representation enables image matching between pairs of images, and thus can be generalized for image matching in other domains. Unfortunately, it is extremely difficult to collect training data of images of actual counterfeit parts; some public repositories exist [11] but they are small in scale. We thus learn our feature representations using a proxy task for which we can readily collect training and testing data: identifying the manufacturer of an IC. This task is useful in and of itself as a means of automatically organizing large-scale collections of ICs. For the counterfeit detection problem, we expect that our dataset, techniques, and learned representations will be useful once suitable training datasets exist, just as models initialized with ImageNet [12] but fine-tuned on smaller-scale, domain-specific datasets have been useful for innumerable applications (e.g., through transfer learning).

In summary, the main contributions of this paper are threefold. First, we explore different variations of Siamese networks and learn an embedding using a combined contrastive and classification loss for the problem of microelectronics imagery analysis. Second, in order to evaluate our approach, we collected and manually annotated a large dataset of 6,387 IC chip images from a diverse set of manufacturers. Finally, we show the effectiveness of our learned embedding on three challenging computer vision tasks in the microelectronics domain: 1) fine-grained retrieval, 2) fine-grained recognition, and 3) image verification.

II. Related Work

Our work is most related to two general threads of research. One is in learning embeddings for image matching, which has become a popular topic in recent years. The other is work specifically targeting recognition and analysis of microelectronic imagery.

A. Representation learning for image matching

Bell et al. [13] proposed a convolutional neural network (CNN) to learn the visual similarity between a product image and its real counterpart in natural images. Household products such as chairs, lightbulbs, tables, etc. are sold on commercial websites and have very different, often idealized appearances compared to real images. Bell et al.'s method learns an embedding that can accurately retrieve product images even across image types. OPNet [14] follows a similar approach to learn a view-manifold of images for the task of novel object recognition. Their method creates a distance metric learned from a CNN. They experiment on a synthetic dataset formed by rendering many views of 3D object models against clean backgrounds. Both methods apply a Siamese network architecture to learn a distance metric on image similarity. Inspired by these papers, we propose a similar distance metric framework to learn the embedding of microelectronic images, and demonstrate the effectiveness of the learned embedding to solve recognition, retrieval, and verification tasks.

Schroff et al. [15] developed FaceNet, a triplet network for face recognition. Their method uses a novel triplet sampling algorithm that improves the learning of the distance metric. Yi et al. [16] developed a deep representation for face images and address both identification and verification. Song et al. [17] learned a deep metric by lifting the structure of distances between a pair of images. They demonstrated that their learned embedding is useful for retrieval on a diverse set of real-world images. Wu et al. [18] showed that the learned deep embedding is affected by how pairs of images are sampled during training time.

In the domain of video analysis, Fan et al. [19] proposed a method for mapping between first-person (egocentric) and third-person camera views by learning a joint embedding space. Xu et al. [20] extended this to segment people common across different views. Both of these papers employed an embedding learning framework to learn common representations from two different camera views.

B. Microelectronics image analysis

In the specific domain of microelectronics image analysis, computer vision techniques have long been used to inspect integrated circuits and other electronic parts for defects [21], [22]. In a supply chain context, Chen et al. [23] highlight some of the challenges and opportunities of using computer vision for improving the security of the microelectronics supply chain. They also investigate integrated circuit image matching and clustering. Wu et al. [24] consider the task of microelectronic component detection (localization and segmentation) in cluttered printed circuit board (PCB) images using a graph embedding network, pointing out that the problem is much harder than one might expect. Reza et al. [25] present techniques for overcoming these challenges. Their paper also conducted preliminary studies of deep learning-based representations for verifying if two IC images correspond to the same part or not; our paper builds on that work. Dhanuskodi et al. [26] explored an alternative approach to IC verification by extracting a "fingerprint" unique to each specific integrated circuit package based on extremely fine-grained analysis of the texture pattern in the plastic package molding. These fingerprints can be enrolled in a database and then verified later even after traversing an untrusted supply chain.

We study the recognition, retrieval, and verification of cropped microelectronic integrated circuit images. The above papers are thus complementary to ours: theirs could be used for detecting and isolating ICs in PCB images, or for tracking individual IC packages, for example, whereas our goal is to perform fine-grained matching and identification of parts without requiring them to be explicitly enrolled at manufacturing time.

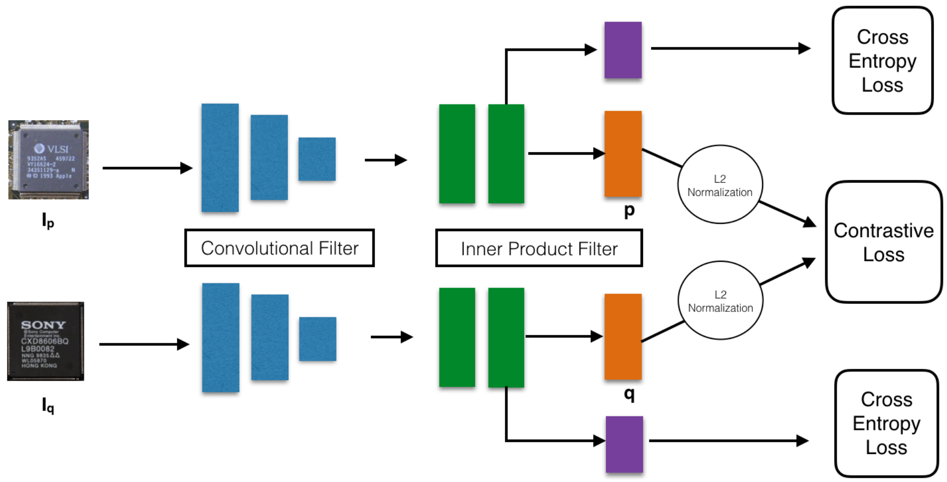

Fig. 2: IC-ChipNet architectural layout. Pairs of images are fed through the convolutional filters (shared weights) to extract feature maps in blue. These feature maps go through the inner-product layer (shared weights) to reach a latent representation (orange). A combination of contrastive and cross-entropy loss allows us to learn the embedding from a large number of input images.

III. Method

To work towards developing a system that can assist microelectronics security applications, we propose a deep neural network for learning an embedding for IC chip images. We are interested in solving three different tasks: retrieval (given a target IC image, finding the most similar IC in a library of other images), recognition (given a target IC image, identify properties of that IC), and verification (given a target IC image, confirm that it is similar to another).

We adopt a Siamese Network framework that can address all of these goals in a unified, principled manner. For IC retrieval, the goal is to find a representation that clusters images of a particular IC type close together while pushing apart others. We achieve this goal by using a *Contrastive* loss function [27]. For IC image recognition, we encourage a representation that can accurately classify using a *Cross Entropy* loss. For verification, the design of the Siamese Network naturally supports this task: comparing two image representations and calculating a measure of their agreement or disagreement.

In more detail, our proposed network is comprised of two copies of the same convolutional neural network (CNN) that share weights between them. Figure 2 shows the layout of the architecture, which we call IC-ChipNet. During training, the Siamese network is presented with pairs of images, one per CNN copy. Let I_p and I_q be two images fed into our network, p and q are the embedded representations from the shared network, and y is an indicator denoting whether the two images come from the same or different classes. IC-ChipNet can be trained to learn an embedding using a loss function consisting of three terms: a contrastive loss between the two images, and two classification losses for each image in the pair.

The contrastive loss is a distance-based loss function that maps semantically similar examples close together in the learned manifold. For a batch of m images, the contrastive loss function is computed as follows,

$$\begin{aligned} L_{contrastive} = \sum_{i \in m} y_i ||p_i - q_i||^2 + \\ +(1-y_i)\max(0, \alpha - ||p_i - q_i||)^2, \end{aligned} \tag{1}$$

where α denotes the *margin* hyper-parameter of the Siamese network.

The cross-entropy losses L_p and L_q encourage the classification capability of the network for the two input images. We use standard *cross-entropy*. The entire network is then trained jointly using the combined loss function,

$$L_{ICChipNet} = L_p + L_{contrastive} + L_q, \tag{2}$$

where L_p comes from the top branch in Figure 2, L_q comes from the bottom branch, and $L_{contrastive}$ is from the middle.

Once the network is trained, the learned embedding for a new image can be found by passing it through either of the two subnetworks and extracting the representation (shown in orange in Figure 2).

IV. Dataset

Deep machine learning requires large-scale training data in order to produce reasonable learned models. Unfortunately, collecting large-scale datasets of counterfeit and defective ICs is inherently difficult, and entities that have these datasets have little incentive to share them with the public. We are aware of one such effort to collect images of defective or counterfeit ICs and make them public, but the dataset is relatively small-scale due to the inherent difficulties of collecting this imagery [11]. Thus we do not restrict ourselves to images of counterfeit ICs nor the problem of distinguishing legitimate

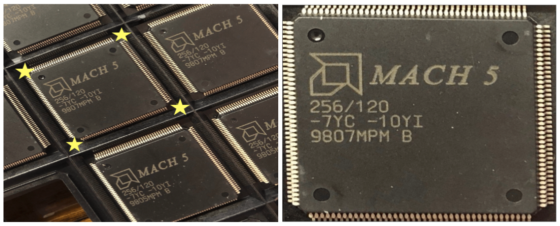

Fig. 3: Annotation process to crop out and rectify IC images. The left shows four manually-annotated points in yellow on a selected IC. The rectified image is shown on the right.

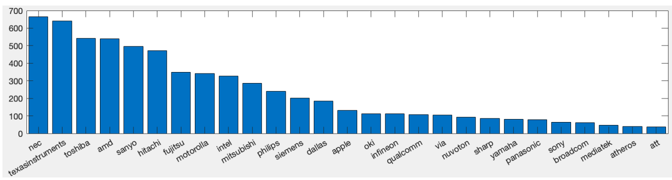

Fig. 4: Frequency of images across different manufacturers in our dataset.

and counterfeit parts, but instead consider proxy problems for which large-scale data can be collected, such as identifying the manufacturer of an IC. Several papers have collected datasets for problems related to computer vision on microelectronics images, such as Kuo et al. [24] and Reza et al [25] who collected printed circuit board (PCB) images from the Internet for the task of PCB component detection, but these were not of sufficient size for learning our task.

We thus collected a dataset of 6,387 high-resolution IC images collected from the Internet. We manually crawled through a variety of websites of electronics retailers, manufacturers, and wholesalers, such as www.ebay.com, www.amazon.com, and www.digchip.com, and collected over 10,000 raw images of ICs. The lighting conditions, presence of noise, watermarks, viewpoint, resolution differences, duplicate images, etc. made it very challenging to use these images directly for further analysis. Some images also contained multiple ICs, or ICs along with other image content. To address these issues, we first manually annotated the quadrilateral region around each IC according to its four corners, cropped it out from the source image, and then applied a perspective transformation to rectify it. Figure 3(left) shows the annotation process for a sample image.

For quality control of the cropped and rectified images, we cross-checked and inspected over multiple iterations and discarded duplicate images. Our final collection contains 6,387 cropped and rectified IC images at various resolutions ranging from hundreds to thousands of pixels. We resized all images to 64x64 pixels for our experiments. We also annotated each IC image with the manufacturer by looking for logos, text, or other markings. In total there were 27 different manufacturers in the dataset, with between about 40 and 650 images per manufacturer. Figure 4 shows the distribution of images across manufacturers.

Split	# Frames
Image for train	4470
Image for validation	642
Image for test	1275
Pairs of images for train	8886
Pairs of images for validation	1230

TABLE I: Statistics of images used in our experiments.

V. Experiments

To evaluate our techniques, we partitioned our dataset of 6,387 images into 70% training images, 10% validation images, and 20% test images (see Table I). For recognition experiments, we used the 27 manufacturer classes described above. In training, we sampled pairs of positive and negative images according to these labels to train our network; a pair is labeled as positive if the two images have the same manufacturer, and negative otherwise. From the train and validation partition of our images, we generated a random collection of training and validation pairs of positive and negative images. To ensure a balance in training, we sampled an equal number of positive and negative pairs in both partitions [17], [18]. Rather than exhaustively generating all permutations of positive pairs within images of a particular manufacturer, we sampled pairs of adjacent images in an ordered list within a particular manufacturer. For each of the positive images in that pair, we randomly selected another negative image from a different manufacturer selected at random. This generated 8,886 pairs of training images and 1,230 pairs of validation images. As is customary, the validation images were used to select the optimal model after training.

A. Network Details

We evaluated our IC-ChipNet using three widely-used convolutional neural networks as the backbone: *AlexNet* [28], *VGG* [29], and *ResNet* [30]. The last layer in all the backbone networks is replaced with an inner-product layer. The dimension of this inner-product layer defines the size of our learned embedding, and in all the experiments it is 4096. Additionally, in our ResNet backbone architecture, an inner-product layer of dimension 1000 is added just before the final inner-product layer. We applied transfer learning by using the pre-trained weights of the backbone networks trained on Imagenet [12] prior to training IC-ChipNet. We trained six variations of IC-ChipNet:

- **IC-ChipNetv1**: All inner-product layers of the backbone *AlexNet* [28] and embedding layer are fine-tuned.
- **IC-ChipNetv2**: All the layers of the backbone *AlexNet* [28] are fine-tuned.
- **IC-ChipNetv3**: All inner-product layers of the backbone *VGG16* [29] and embedding layer are fine-tuned.
- **IC-ChipNetv4**: All the layers of the backbone *VGG16* [29] are fine-tuned.
- **IC-ChipNetv5**: All the layers of the backbone *ResNet-50* [30] are fine-tuned.

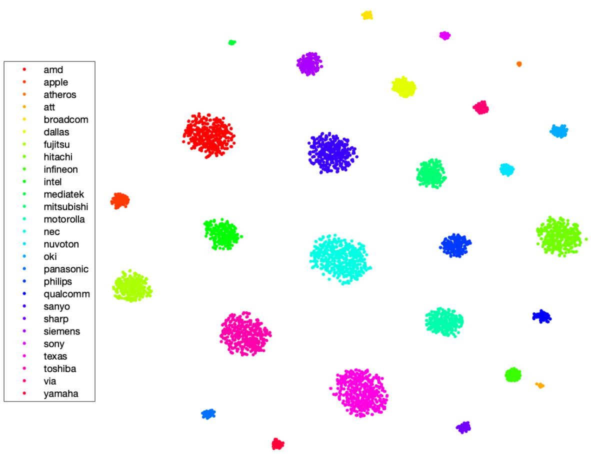

Fig. 5: Visualization of the learned embedding of the training instances for IC-ChipNetv5 (best viewed in color). The embedding is visualized using t-SNE [32].

- **IC-ChipNetv6**: All the layers of the backbone *ResNet-101* [30] are fine-tuned.

We implemented our method in PyTorch. Our networks are trained in mini-batches using the Adam [31] optimizer using a mini-batch of size $m = 4$ in all experiments. The maximum epoch used in our experiments is 300. We tried two different values of the margin hyper-parameter for the contrastive loss function, $\alpha = \{0.447, 1.0\}$, and finally selected a value of 0.447 for all our experiments. We also tried different values for learning rates and selected the optimum. For all the experiments, an initial learning rate of $lr = \{10^{-6}\}$ was used.

B. Baselines

To put our results into context, we compared against three baseline methods:

- **AlexNet:** We utilized a pre-trained AlexNet [28] model as our first baseline. We used the "fc7" feature — the last inner-product layer before the classification layer — as the representation. The dimension of this feature is 4096.
- **Variational Autoencoder (VAE):** We trained our VAE from scratch using the images in the training set for the task of reconstruction [33]. The latent representation of VAE was used as the final representation.
- **Siamese Network with Contrastive Loss:** We also compared against three other baseline Siamese Networks and trained them with only contrastive losses [27]. The backbone networks used in this setting were AlexNet [28], VGG-16 [29], and ResNet-50 [30]. We refer to these three baselines as *Siamese-baseline1, Siamese-baseline2*, and *Siamese-baseline3* respectively.

C. Results

The learned embeddings on the training set are visualized in Figure 5 using t-SNE [32]. Notice how well the embeddings are clustered around each manufacturer's image group,

Method	Accuracy	Recall@1	Recall@7
AlexNet [28]	60.08%	60.08%	83.06%
VAE [33]	19.84%	9.41%	43.06%
Siamese-baseline1	31.92%	24.24%	60.31%
Siamese-baseline2	21.80%	17.18%	57.33%
Siamese-baseline3	37.88%	30.27%	63.61%
IC-ChipNetv1	73.02%	73.02%	83.22%
IC-ChipNetv2	78.35%	76.94%	83.69%
IC-ChipNetv3	72.39%	70.59%	74.67%
IC-ChipNetv4	82.75%	81.49%	83.84%
IC-ChipNetv5	**83.69%**	**83.22%**	**86.59%**
IC-ChipNetv6	80.94%	80.24%	85.57%

TABLE II: Evaluation for the IC image recognition task (column 2) and retrieval task (column 3 and 4) using the proposed IC-ChipNets and other baselines (top five rows).

thus suggesting that the network has learned a reasonable representation. We evaluated the performance of our learned embedding in solving three different tasks: (1) retrieval of images of the same manufacturer, (2) recognition of the manufacturer, and (3) verification of whether two images have the same manufacturer.

1) Retrieval: In this experiment we investigated the effectiveness of the embedding for finding similar IC images. Figure 6 shows some sample retrieved images from the training set, when we use a test image as a query. Additional retrieval results are shown in Figures 11 and 12. For a quantitative analysis, we report the retrieval performance in Table II (columns 3 and 4) using the Recall@K metric. Based on the learned embedding, we first retrieve the K closest images from the training set. For each test image, the recall will be 1 if an image from the same manufacturer appears in the pool of closest K, and 0 otherwise. Then, we average this value across all the test images to find the overall Recall@K number. Similarity in the ranking is based on cosine distance.

The best performance was achieved by IC-ChipNetv5 (with a ResNet-50 [30] backbone) with a Recall@7 value of 86.59%. The progression of the recall metric for various K values is shown in Figure 7. The best performing IC-ChipNetv5 is shown with a blue dashed curve. As the neighbor count (value of K) increases, so does the chance of randomly retrieving images of a similar type. Recall@K is a weaker metric than accuracy in this regard. In order to correctly classify an example, the majority of the retrieved samples have to come into an agreement. Recall@K, on the other hand, will count a retrieval successful if any one of the retrieved images is from the same manufacturer.

2) Recognition: To measure recognition performance, we compute accuracy, which is the total number of correctly predicted images divided by the total number of images in the test set. For recognition, we retrieved the top@K closest images to the target from the training set using the cosine distance, and if the majority of the retrieved images have the same label as the target's ground truth, then it is considered correct.

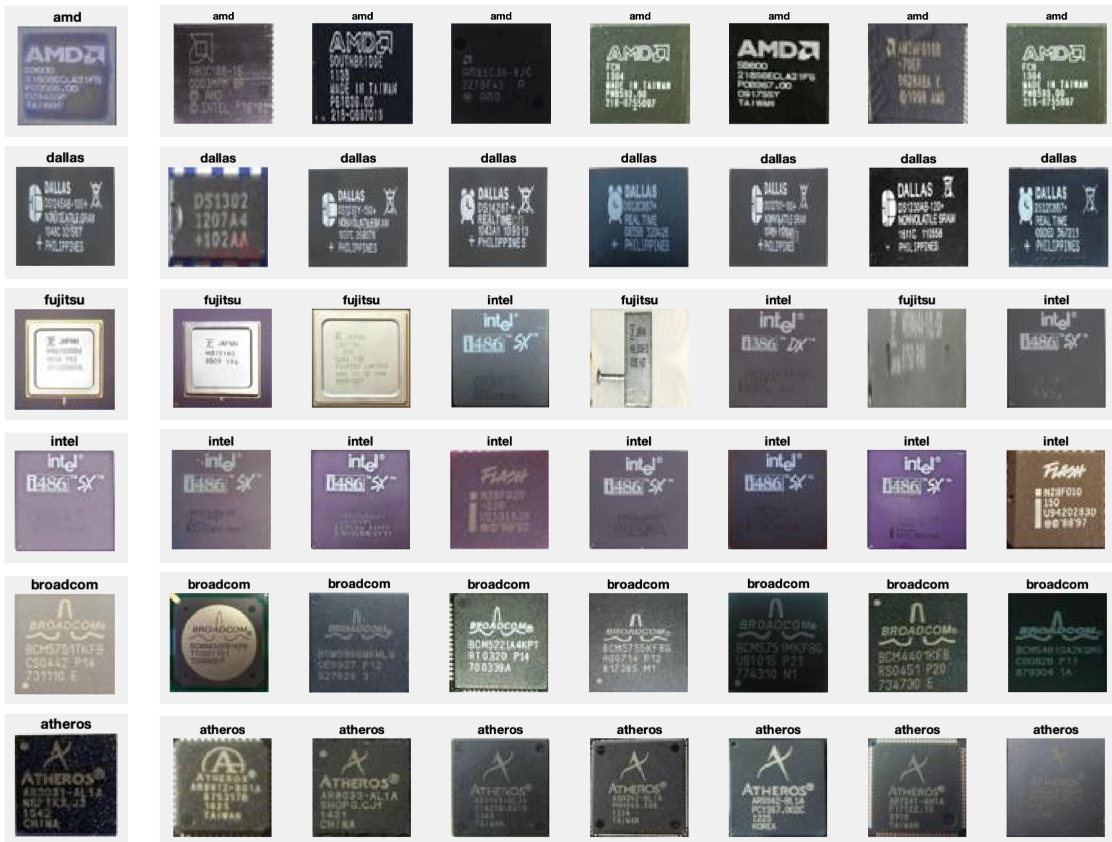

Fig. 6: The retrieved top@7 examples from the training set. In each row, the query image is shown in first column, and the seven images to the right show the query results.

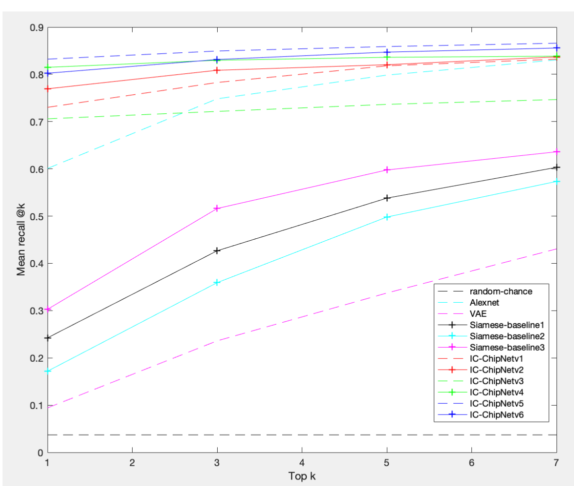

Fig. 7: Recall@K curve for the retrieval task (best viewed in color). The learned embedding retrieves images from the same manufacturers. As the K value increases, the baseline method catches up.

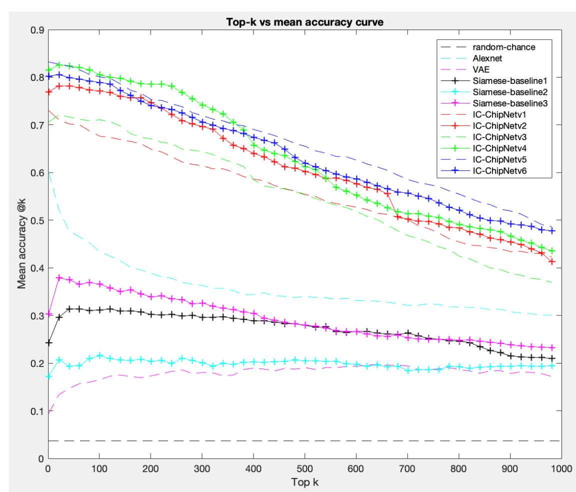

Fig. 8: Accuracy curve for the recognition task for various values of K (the number of nearest neighbors).

The performances for the recognition task are reported in Table II (column 2). Each entry denotes the best accuracy found for any value of K in the nearest-neighbor search. The overall curve for all possible values of K is shown in Figure 8. The best performing model on the test set is *IC-ChipNetv5* (with ResNet50 [30] backbone) with 83.69% accuracy. It

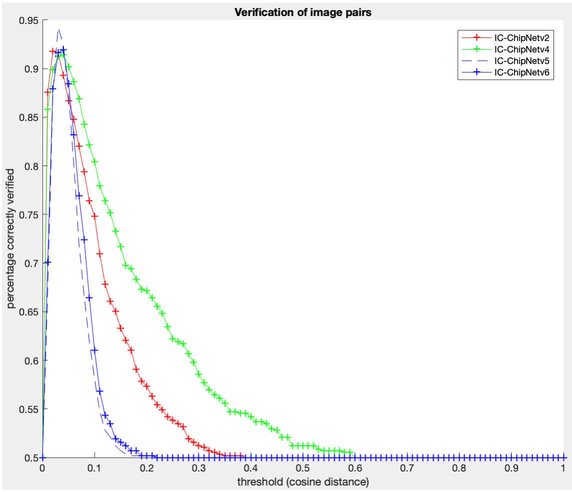

Fig. 9: Accuracy vs threshold curves for the verification task (best viewed in color).

Fig. 10: Sample images from our verification experiment. In each row, the left image is the source from which a pair of IC images were cropped. The cropped image pairs are shown in the second and third columns.

performs better than *IC-ChipNetv2* (AlexNet [28] backbone) by 5.34%. IC-ChipNetv4 (with VGG16 [29] backbone) also performs well with an accuracy of 82.75%.

In general, we observed a trend of improving performance with deeper networks. Relearning all the layers of the backbone helps to improve the performance. Our VAE baseline has poor performance. We analyzed the reconstruction output of this network on the test set; although the reconstruction nicely finds the overall appearance of an image, it fails to reconstruct the finer details. Our findings are in agreement with the observations of Hou et al. [34].

3) Verification: Finally we conducted an experiment to test how useful our learned embedding is in deciding whether two IC images are from the same manufacturer or not. We set aside a separate subset of 286 source images consisting of multiple copies of an IC chip. During our dataset preparation, we only

Method	Best Accuracy	Threshold
IC-ChipNetv2	91.78%	0.02
IC-ChipNetv4	91.43%	0.04
IC-ChipNetv5	**94.23%**	0.03
IC-ChipNetv6	91.96%	0.04

TABLE III: Verification experiment results. The best accuracy and corresponding threshold of correctly predicting a pair of images are reported in 2nd and 3rd column respectively.

took one instance of that IC image at random if multiple copies were present. A few sample source images are shown in the first column of Figure 10. The other two columns in each row show two copies of that IC that were cropped at random in order to conduct our verification experiment.

We prepared 286 pairs of positive instances — pairs of ICs that have the same manufacturer — for the verification experiment. For each pair of positive instances, we randomly picked another IC image from a different manufacturer to create a new negative pair instance, resulting in an additional 286 negative pairs. Thus our verification test set contains in total 572 pairs of IC images from 16 different manufacturers.[1] We extracted the embedding from a subset of better performing models (*IC-ChipNetv2, IC-ChipNetv4, IC-ChipNetv5, IC-ChipNetv6*) for each of these pairs of images, then computed cosine distance between each pair. If two IC images are very similar then the distance between them should be very small.

We computed the verification accuracy as a function of different threshold values as shown in Figure 9. *IC-ChipNetv5* performed the best, achieving an accuracy of 94.23% at threshold 0.03. As the verification-accuracy curve points out, it was sufficient to pick a small threshold to correctly label two similar-looking IC images. This experiment also confirms the effectiveness of the learned embedding for the IC image verification task. The best accuracy and the corresponding threshold are also reported in Table III.

VI. Conclusion

In this paper, we learned an embedding using a deep metric learning approach from a pair of IC images. Our learned embedding was demonstrated to be very effective for the recognition and retrieval of IC images. We also demonstrated that an IC image can be compared with a known reference image, to verify that it is the same device. Finally, we contributed a large repository of 6,387 IC chip images to advance research in problems related to microelectronic imagery. Our method also establishes baselines for the three computer vision tasks for microelectronic image analysis.

VII. Acknowledgments

This research was funded by the Indiana Innovation Institute (IN3). We also would like to thank following Indiana University undergraduate students for assisting us in image collection and annotation: Kristopher Jung, Yongming Fan, Zunaeed Salahuddin, John Kaefer, Tiancong Zhao, and He He.

[1] *amd, att, broadcom, dallas, hitachi, infineon, intel, mitsubishi, motorolla, oki, philips, sanyo, sharp, texas instruments, toshiba*, and *via*

Fig. 11: The retrieved top@7 examples from training set. In each row, the leftmost image is the query, and the seven images to the right are the retrieved ICs.

Fig. 12: The retrieved top@7 examples from training set. In each row, the leftmost image is the query, and the seven images to the right are the retrieved ICs.

REFERENCES

[1] "The big hack: How China used a tiny chip to infiltrate U.S. companies," *Bloomberg*, Oct 4, 2018.

[2] S. Greengard, "Are computer chips the new security threat?" *Communications of the ACM*, February 2017.

[3] M. M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit integrated circuits*. Springer, 2015.

[4] N. Seidle, "ATmega Slugs," https://www.sparkfun.com/news/350, April 21, 2010.

[5] ——, "Revisiting the Counterfeit ATMega328s," https://www.sparkfun.com/news/364, May 17, 2010.

[6] ——, "Dear ON Semiconductor," https://www.sparkfun.com/news/384, June 25, 2010.

[7] ——, "Fake ICs Identified," https://www.sparkfun.com/news/395, July 12, 2010.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2016.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2016.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] N. Asadizanjani, S. Gattigowda, N. Dunn, M. Tehranipoor, and D. Forte, "A database for counterfeit electronics and automatic defect detection based on image processing and machine learning," 2016.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," in *ACM Transaction on Graphics (SIGGRAPH)*, 2015.

[14] X. Lin, H. Wang, Z. Li, Y. Zhang, A. Yuille, and T. Lee, "Transfer of view-manifold learning to similarity perception of novel objects," in *IEEE Conference on Computer Vision and Pattern Recognition (ICLR)*, 2017.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] S. Yi, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[17] H. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] C. Wu, R. Manmatha, A. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," in *International Conference of Computer Vision (ICCV)*, 2017.

[19] C. Fan, J. Lee, M. Xu, K. Singh, Y. Lee, D. Crandall, and M. Ryoo, "Identifying first-person camera wearers in third-person videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] M. Xu, C. Fan, Y. Wang, M. Ryoo, and D. Crandall, "Joint person segmentation and identification in synchronized first- and third-person videos," in *European Conference on Computer Vision (ECCV)*, 2018.

[21] M. Baird, "EYESEE: a machine vision system for inspection of integrated circuit chips," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985.

[22] C. Wang, B. C. Jiang, J. Lin, and C. Chu, "Machine vision-based defect detection in ic images using the partial information correlation coefficient," *IEEE Transactions on Semiconductor Manufacturing*, vol. 26, no. 3, 2013.

[23] Z. Chen, T. Wanyan, R. Rao, B. Cutilli, J. Sowinski, D. Crandall, and R. Templeman, "Addressing supply chain risks of microelectronic devices through computer vision," in *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2017.

[24] C.-W. Kuo, J. Ashmore, D. Huggins, and Z. Kira, "Data-efficient graph embedding learning for pcb component detection," in *Winter Conference of Computer Vision (WACV)*, 2019.

[25] M. A. Reza, Z. Chen, and D. J. Crandall, "Deep neural network-based detection and verification of microelectronic images," *Journal of Hardware and Systems Security*, vol. 4, pp. 44–54, 2020.

[26] S. N. Dhanuskodi, X. Li, and D. E. Holcomb, "COUNTERFOIL: verifying provenance of integrated circuits using intrinsic package fingerprints and inexpensive cameras," in *29th USENIX Security Symposium*, 2020.

[27] R. Hadsell, S. Chopra, and Y. LeCun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems (NIPS)*, 2012.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representation (ICLR)*, 2015.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *CoRL*, 2014.

[32] L. Maaten and J. Hinton, "Dimensionality reduction by learning an invariant mapping," in *Journal of Machine Learning Research*, 2008.

[33] D. Kingma and M. Weilling, "Auto-encoding variational bayes," in *arXiv*, 2013.

[34] X. Hou, L. Shen, K. Sun, and G. Qiu, "Deep feature consistent variational autoencoder," in *Winter Conference of Computer Vision (WACV)*, 2017.