On Identification and Retrieval of Near-Duplicate Biological Images: a New Dataset and Protocol

T.E. Koker[§], S.S. Chintapalli[§], S. Wang, B.A. Talbot, D. Wainstock, M. Cicconet, and M.C. Walsh
Harvard Medical School
Email: teddy.koker@gmail.com, sc4425@columbia.edu, {san_wang, blake_talbot, daniel_wainstock, marcelo_cicconet, mary_walsh}@hms.harvard.edu
[A version of this article appears in ICPR 2020.]

***Abstract*—Manipulation and re-use of images in scientific publications is a growing issue, not only for biomedical publishers, but also for the research community in general. In this work we introduce BINDER – Bio-Image Near-Duplicate Examples Repository, a novel dataset to help researchers develop, train, and test models to detect same-source biomedical images. BINDER contains 7,490 unique image patches for model training, 1,821 same-size patch duplicates for validation and testing, and 868 different-size image/patch pairs for image retrieval validation and testing. Except for the training set, patches already contain manipulations including rotation, translation, scale, perspective transform, contrast adjustment and/or compression artifacts. We further use the dataset to demonstrate how novel adaptations of existing image retrieval and metric learning models can be applied to achieve high-accuracy inference results, creating a baseline for future work. In aggregate, we thus present a supervised protocol for near-duplicate image identification and retrieval without any "real-world" training example. Our dataset and source code are available at** hms-idac.github.io/BINDER**.**

I. Introduction

In this paper, we outline the need and intention behind generating a large dataset of synthetically manipulated biological images that emulate the examples of same-source image manipulations found in the scientific literature. In addition, we adapt and evaluate modern methods in image retrieval and metric learning to address the problem of detecting biomedical image data manipulation and re-use. Using a large corpus of synthetically manipulated images (derived *online* from a set of open-source available biological images) we train a selection of metric learning models minimizing Euclidean distance between same-source image pairs.

Our dataset of synthetically manipulated biological images consists of training, test, and validation subsets. The training set contains 7,490 unique image patches (2-D gray-scaled images of size 256x256), that have a series of manipulations (or deformations) introduced during the training process to generate unique samples that are then center-cropped to 128x128. The test and validation sets contain 799 and 1022 image pairs (source and source-manipulated 2-D gray-scaled images of size 128x128 respectively). Importantly, we fix the manipulations for the validation and test sets in order to create a benchmark dataset facilitating reproducibile, replicable and robust comparative assessments for future methods [13], [21].

[§]Equal contribution.

Finally, we have created supplemental validation and test sets where, for each pair, one image is of size 128x128 and the other is of arbitrary larger size, with the smaller image being a manipulation of a crop of the larger image.

II. Related Work

In the recent past, deep learning and computer vision communities have proposed several methods and benchmark datasets to tackle the problem of near-duplicate image detection. [6] gathered a dataset of near-duplicate images from the MIR-Flickr image database of 1 million images [9]. In their work, they primarily focused on identifying "identical" near-duplicates (generated through some transformation from the same source) but also gathered non-identical near-duplicates (images that share scenes and objects). In [11], the authors present an algorithm to identify identical near-duplicates, their method uses Lowe's difference of gaussian (DOG) detector to identify keypoints from images followed by PCA-SIFT to compute local descriptors that are invariant to certain transformations. They employ locality sensitive hashing to index the descriptors and retrieve near-duplicates. [12] presents a method that uses cluster seed generation and cluster-growing through min-hash filters to identify non-identical near-duplicates from a database of 1 billion images. In [17], the authors compare the performance of state-of-the-art deep learning models for near-duplicate image detection on multiple datasets encompassing images of indoor/outdoor scenes, objects, and people.

The term "near-duplicate" has been used ambiguously in the past to describe 1) images that share the same objects and scene, and 2) images that are copies of the same digital source that were manipulated/transformed. In our paper we concentrate on the latter: our primary goal is to establish a benchmark dataset of images that were duplicated from the same source and manipulated, and utilize the dataset to study the performance of modern deep learning methods for this problem. We focus on microscopic images due to our closer connection to this field.

According to [2], [3], inappropriate image duplication in biomedical literature is a growing challenge for the scientific community. The authors estimated that about 35,000 papers may contain retraction-worthy instances of image duplication. The prevalence of duplicative images in biomedical journals is a concern with respect to research integrity as it raises

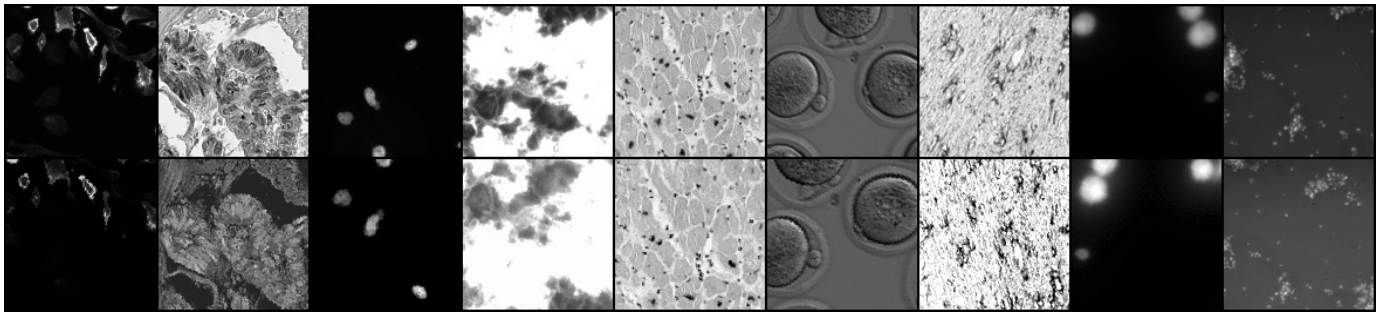

Fig. 1. Examples of duplicate pairs from each of the 9 classes used for training. Row 1 displays original patches from various sources; Row 2 displays deformations of the respective patches in Row 1.

questions about the reliability and validity of the published work. These instances of possible research misconduct through image duplication within the biomedical literature have been previously studied and efforts have been made to develop methods that can be employed to detect duplicative images.

In [1], [22], the authors propose computer-assisted methods to detect biomedical images that are same source-manipulated. In [1], SIFT (scale-invariant feature transform) [15] is used for feature extraction followed by a feature matching algorithm to identify manipulated images, a patch-wise classifier is then used as the final step to predict if image patches show biological images. In [22], a patch-based feature extractor is applied to residual images (residual images are computed as the difference between the original and filtered image), and a detector is built to detect outlier patches. These methods use explicit feature extraction techniques, hence are limited to a narrow range of image manipulations.

It is known that in order to study the performance of deep neural networks a relatively large dataset is desired. But for our problem access to such dataset from "real-world" cases is difficult (mostly due to legal constraints). There is at least one publicly available related dataset, MFND IND [6], of 1958 same-source image clusters, but these are not biological images. Thus we endeavored to build our own dataset.

III. BINDER: Bio-Image Near-Duplicate Examples Repository

Published images that included confirmed duplication (confirmed through institutional and/or regulatory processes), including retracted and or corrected data, are not identifiable and available at a volume that would be required to train a high capacity neural network. Additionally, to create an openly available database of published images that were flagged as problematic, the authors and publishers must be willing to share their data. As a measure to overcome said limitations, we created our dataset using synthetic manipulations.

We identified the most common manipulations within a small set of duplicative images (126 image pairs) found in peer-reviewed publications[1]. However, this dataset (let us call it RWE – for real-world examples) is comprised of images that may be subject to copyright protections that prevent us from sharing more broadly. RWE has a substantial representation of manipulations and alterations that we identified as often found within the broader public literature, including: rotation, translation, scale, or perspective transform; gamma correction, brightness and contrast adjustment; and jpeg compression artifacts. We therefore translated these categories into operations within our framework which may be applied to any biological image for creation of a shareable dataset.

We gathered biological (microscopy) images from the following open-source public repositories: NYU Mouse Embryo Tracking Database[2] (METD), the Broad Bioimage Benchmark Collection[3] (BBBC), the Adiposoft Image Dataset[4] (AID), and the Open Microscopy Image Data Resource[5] (IDR). The images encompass 17 classes/categories, from various cell types and model organisms, including C. elegans, adipocytes, human histology, mouse nuclei, and mouse embryo, captured through different microscopes. Figure 1 contains representative patches for each of the 9 categories in the training set; Row 1 demonstrates examples of original patches from various sources and Row 2 displays deformations (series of transforms of various types) of respective patches in Row 1.

To create BINDER we split the microscopy images class-wise into training, test, and validation sets (17 categories split into 9, 4, and 4 classes respectively). Images were tiled in 256x256-pixel non-overlapping patches. As shown in Figure 2 (a), for each patch in the validation and test sets, manipulations shown in Table I were applied to create a duplicate patch. Both the original and duplicate patch were then center-cropped to 128x128 pixels to remove any border artifacts caused by the manipulations. Finally, we manually removed any image pairs without meaningful content, i.e. images that were blank or just noise. The training set was left as 256x256-pixel patches so the same manipulation procedure could be applied online during training.

The choice of patch size was made based on image content

[1]These were obtained through PubPeer (https://pubpeer.com) and/or Retraction Watch (http://retractionwatch.com). We will not be able to distribute this dataset due to legal restrictions.

[2]http://celltracking.bio.nyu.edu/

[3]https://data.broadinstitute.org/bbbc/, datasets BBBC 002, 010, 015, 016, 019, 039, 041, and 042

[4]https://imagej.net/Adiposoft

[5]https://idr.openmicroscopy.org/

TABLE I
SYNTHETIC MANIPULATIONS

Manipulation	Procedure
Vertical Flip	$P = 0.5$
Horizontal Flip	$P = 0.5$
Color Invert	$P = 0.1$
Perspective	Corners perturbed $U(-20, 20)$ pixels
Scale	$U(0.75, 1.25)$
Rotation	$U(-20, 20)$ degrees
x and y Translation	$U(-10, 10)$ pixels each
Gamma Correction	$U(0.5, 1.5)$
Brightness	$U(0.9, 1.1)$[1]
Contrast	$U(0.5, 1.5)$[1]
JPEG Compression	$P = 0.1$, quality $= 50$

P signifies probability of manipulation being performed, U represents the range of random uniform distribution in which manipulation will be applied.
[1] See https://pillow.readthedocs.io/en/stable/reference/ImageEnhance.html for the meaning of parameters.

and computational capacity. Presumably the models would work better on higher resolution, but training and inference would take longer. The choice of grayscale (single-channel) images was done because this is the most general case in biology – while in images from consumer cameras it makes sense to use RGB (3 channels), in biology there is no standard number of channels beyond 1.

IV. MODELS

We frame the task of duplicate image detection as one of image retrieval. Given a query image and potential duplicate images, a model computes a global descriptor (an *embedding*) for each image. An image is decided to be a duplicate to the query image if the euclidean distance between their descriptors is less than a certain threshold, t. Figure 2 illustrates the components of our model.

First, following [18], we investigate two models, each consisting of a pre-trained feature extractor (VGG19[19] and ResNet50[8], respectively) followed by a generalized-mean (GeM) pooling layer, a fully-connected layer, and lastly L_2 normalization. The models are trained to minimize triplet loss over the training set – Figure 2 (c). In previous end-to-end image retrieval works [18], [7], negative pairs are chosen by mining the most similar non-matching images over the entire training set. This procedure is important to ensure non-trivial negative cases but necessitates access to all potential pairs in the training set. In order to obtain hard negative cases (i.e. non-duplicate images that the model thinks are similar) over a dataset in which pairs are generated online, we adopt the use of hardest-in-batch sampling [16]. That is, for each batch $\mathcal{X} = (A_i, B_i)_{i=1..n}$, where A_i and B_i are duplicate images with output embeddings $(a_i, b_i)_{i=1..n}$, we compute the triplet margin loss as:

$$L = \frac{1}{n} \sum_{i=1..n} \max(0, d(a_i, b_i) - \min(d(a_i, b_j), d(b_i, a_k)) + \alpha) \tag{1}$$

where $d(x, y) = \|x - y\|^2$; b_j and a_k are the closest non-duplicates of a_i and b_i, respectively, which are found using the pairwise distance matrix D, where $D_{ij} = d(a_i, b_j)$ (see Figure 2 (b)).

As illustrated in Figure 2 (d), positive image pairs A_i and B_i are fed into the siamese CNN model and their closest non-duplicates (hardest negatives) are selected from within the batch during training. In the triplet loss, α is a hyper-parameter representing the margin between positive and negative pairs. We use a value of $\alpha = 1.0$ for all experiments. The loss function simultaneously rewards the network for maximizing the distance between negative pairs, and minimizing the distance between positive pairs. Hence during training, the model learns a distance metric that places duplicate images close to each other, and non-duplicate images farther apart (Figure 2 (c)).

The models were pre-trained on COCO [14] with an output embedding size of 2048, and a batch size of $n = 256$ image pairs, generated online with synthetic manipulations shown in Table I. In order to maximize batch size given limited memory budget, we use gradient checkpointing as in [5]. Pre-training is done for 75 epochs, with a learning rate of 0.1. This process took less than one day on a single Nvidia 1080 Ti GPU. Pre-training was necessary to prevent overfitting on the biological training set.

Next, each model was fine-tuned on the biological training set (BINDER) using the same hyper-parameters. Model weights with the lowest validation loss were saved for testing.

The third model we analyze is based on [20], [10]. To extract features from the input data, we construct a shallow convolutional siamese autoencoder model also pre-trained on COCO [14]. A siamese autoencoder is made of two autoencoders, where the encoders (E_1 and E_2) share weights and the decoders (D_1 and D_2) have independent weights. The encoder reduces the higher-dimensional input to a lower-dimensional representation while preserving the most important features that the decoder reconstructs back into the original input. Based on this observation, the encoder extracts the lower-dimensional representations ($E_1(A_i)$ and $E_2(B_i)$) from the input pairs (A_i, B_i). The representations are then downsampled using max and average pool layers and the resultant vectors are concatenated ($output_1 = [max_{pool}(E_1(A_i)), average_{pool}(E_1(A_i))]$ and $output_2 = [max_{pool}(E_2(B_i)), average_{pool}(E_2(B_i))]$). The concatenated vectors are passed through fully connected layers with shared weights and the network outputs the resultant embeddings (a_i and b_i respectively of size 256).

This model is pre-trained on COCO for 15 epochs (since the network is relatively shallow) with a learning rate of 0.0001, then fine-tuned on BINDER. For model stability, a batch size of $n = 64$ was selected for pre-training, and $n = 16$ for fine tuning. The network is trained by minimizing the triplet

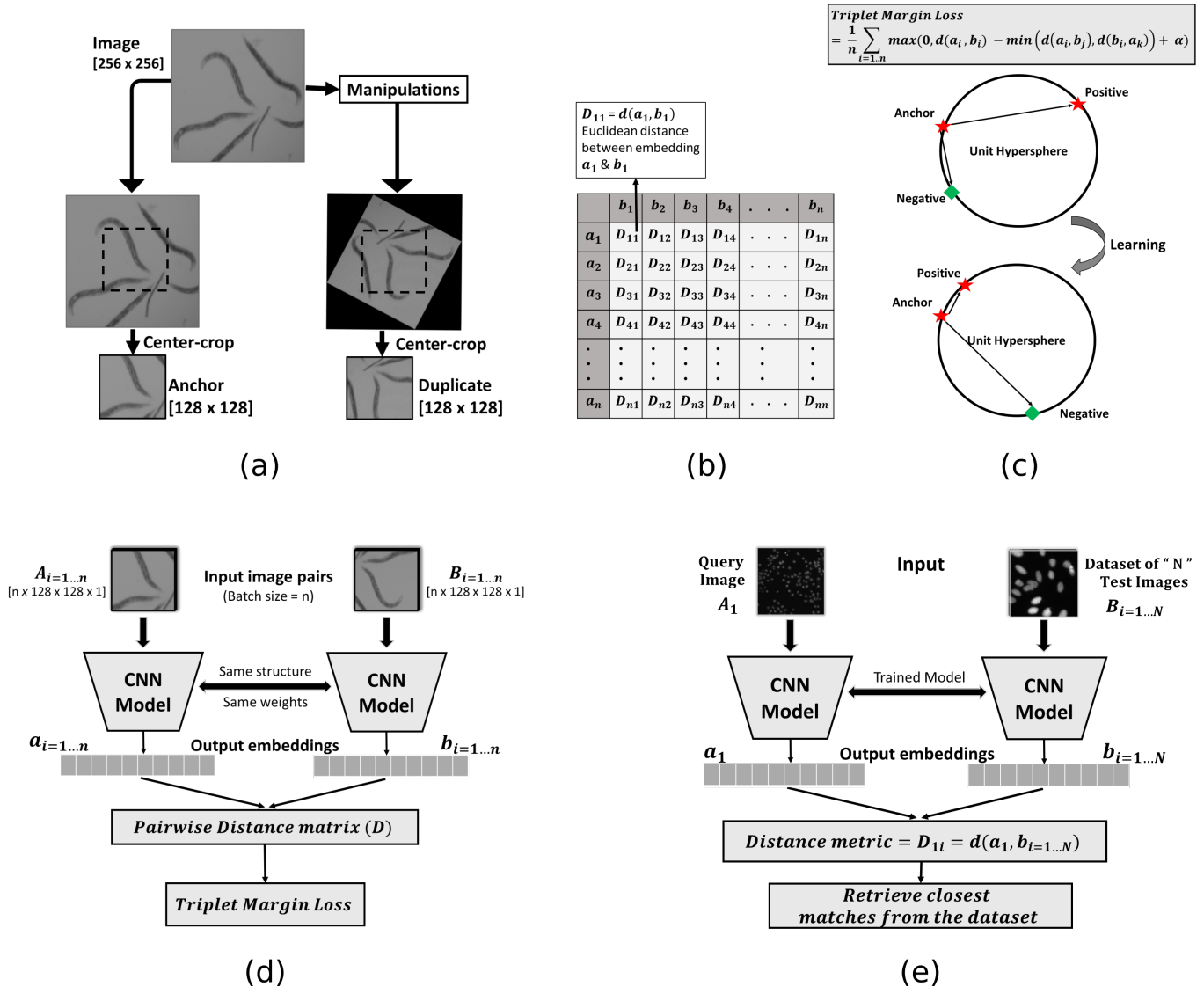

Fig. 2. Model diagrams. (a) Synthetic image manipulation for duplicate pair generation. Random manipulations (see Table I) are applied to generate duplicate images. (b) Pairwise distance matrix as used during training. (c) Triplet margin loss minimizes the distance between positive pairs and maximizes the distance between negative pairs during training. (d) Model training diagram. (e) Model testing/deploying diagram.

margin loss of Equation (1), using random sampling followed by the same hardest-in-batch negative sampling method.

We experimented with a range of architecture designs and hyper-parameters, but did not perform a comprehensive search for the best model since the goal here was to provide baselines from a few inference and generative architectures, as a starting point for other researchers.

V. Evaluation

For evaluation of our models, we first use an approach similar to that of [17]. For each duplicate image pair, we randomly select one to be the query image. We then compose negative and positive pairs:

- Positive pairs are composed of the query image and its duplicate.
- Negative pairs are composed of the same query image and the closest non-duplicate image within the dataset. Since these pairs depend on the distance between images, negative pairs must be composed for each model.

We then quantify the models' ability to retrieve duplicate image pairs by measuring the area under the ROC curve (AUC) [4]. This quantification metric will approximate the models performance over larger datasets, in which the number of non-duplicate pairs will likely be exponentially larger than the number of duplicate pairs [17].

Figures 4 and 5 illustrate examples of closest non-duplicate and farthest duplicate image pairs, respectively.

We do additionally evaluate our model in a similar manner, using randomly selected negative pairs as a comparative baseline assessment. This will indicate a models ability to differentiate random pairs of non-duplicate images, which

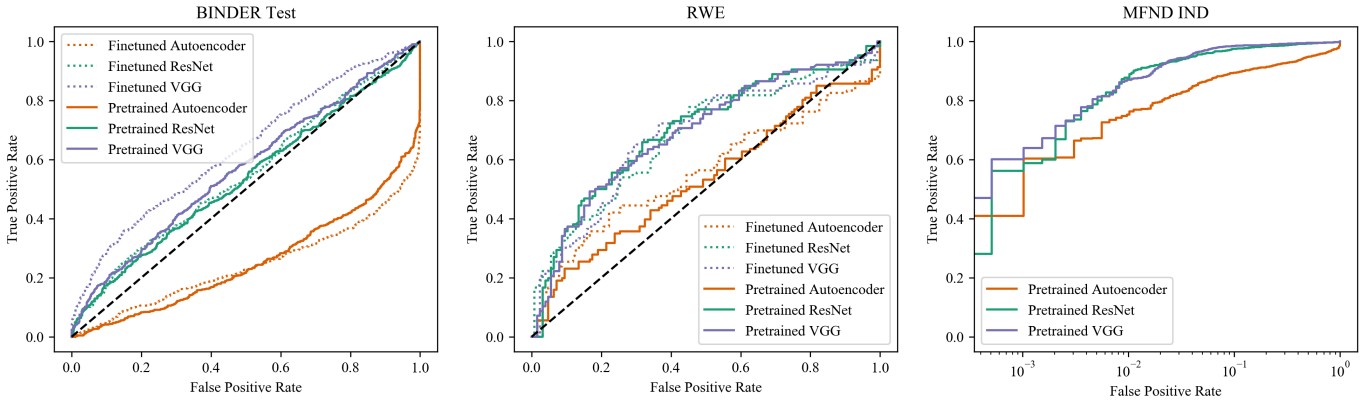

Fig. 3. ROC curve for one run on each dataset, using hard negative mining. Log scaling is used for MFND IND. Threshold is sampled in $[0, 2]$.

should be very high given the trivial nature of most randomly selected pairs.

Each model is evaluated over the following datasets:

1) BINDER test set (799 image pairs).
2) RWE (126 image pairs).
3) MFND IND (1958 image clusters, pairs randomly selected for the few clusters with > 2 images).

Since MFND IND includes images of scenes and objects, it is not evaluated with models fine-tuned on BINDER. All images are converted (if needed) to grayscale 128x128 pixel patches before testing. The model deployment pipeline is illustrated in Figure 2 (e).

VI. Results

Figure 3 shows the models' performances while using hard negative mining on BINDER test, RWE, and MFND IND datasets. The AUC scores for each models' performance, while using hard negative mining and random selection, are listed in Table II. After comparing the AUC scores across all models and datasets, we observe that the AUC scores for random negative sampling are consistently higher than the AUC scores for hardest negative sampling. In Table II, the AUC scores for random negative sampling are in the range 0.92 - 1.00 across all models, whereas for hardest negative sampling the AUC scores are in the range of 0.25 - 0.99. This observation aligns with our initial prediction: since randomly selected negative pairs tend to be easier to distinguish, they result in higher AUC scores. In comparison, hardest negative sampling selects image pairs that may share a lot of features thereby affecting the models' ability to distinguish between non-duplicates. It is worth noting that with hard sampling a model assigning random embeddings to images will achieve an AUC that approaches 0, as there will nearly always be an embedding closer in distance to the query embedding than the actual duplicate. Given that the MFND IND image pairs are all much more unique in nature, the AUC score is comparatively high.

As is evident from Figure 3, the VGG19 + GeM model outperforms the two other models (see purple lines, both solid and dashed, on all plots). In addition, we observe that fine-tuning on the field specific dataset (BINDER) does indeed yield improved results. As shown in Table II, the VGG19 + GEM model's AUC score with hardest negative sampling improved from 0.58 to 0.63.

TABLE II
Model Benchmarks

Model	Fine-tune	BINDER Test	RWE	MFND IND.
Autoencoder	None	0.25/0.92	0.53/0.95	0.935/0.999
Autoencoder	Bio	0.25/0.93	0.55/0.96	
RN50+GeM	None	0.54/0.99	0.70/0.98	0.985/1.000
RN50+GeM	Bio	0.55/0.99	0.68/0.98	
VGG19+GeM	None	0.58/0.99	0.69/0.99	0.988/1.000
VGG19+GeM	Bio	**0.63**/0.99	0.69/0.98	

Area under the ROC curve for each dataset tested with [hardest/random] negative sampling. Each result is the median of 5 runs.

Finally, as a visual example of how the models make calls on image identities, Figure 4 demonstrated how one of the models (VGG19 model shown) differentiate non-matching (but similar) image data. For example, in Figure 4, compare Column 1 [input images] vs Column 2 [non-matching (but visually similar) negative controls]. Alternatively in Figure 5, the model demonstrates its ability to identify duplicated images that may be altered or modified. For example, in Figure 5, compare Column 1 [input images] vs Column 2 [matching (but modified) matched images].

VII. Conclusion

Our primary contribution is towards creating an assessment platform to identify and metrically assess image duplications with and without modifications in the recording or reporting of scientific research. We have also established a methodology (protocol) for generating a standard dataset of synthetically manipulated biological images; both for utilization with our models of metric assessment and for establishing a benchmark to evaluating and validating similar work in the community. This includes generating a static image reference set (BINDER) that we are making available with our code via our project page: hms-idac.github.io/BINDER/

We tested the performance of three deep neural networks that were pretrained on COCO and fine-tuned on BINDER's

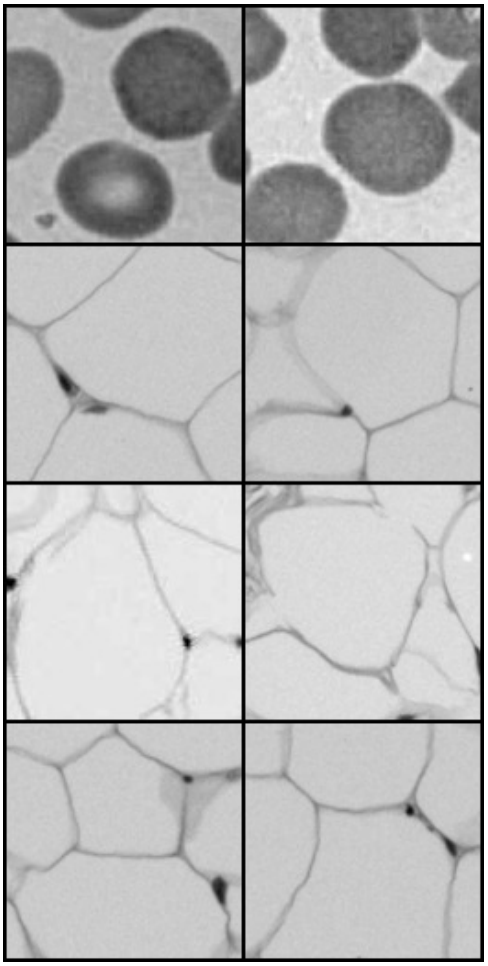

Fig. 4. Closest non-duplicate images pairs in the test set, using VGG19 model. Query image is on the left (Column 1), and examples of model selection of their closest matching associated (but different) images are on the right (Column 2).

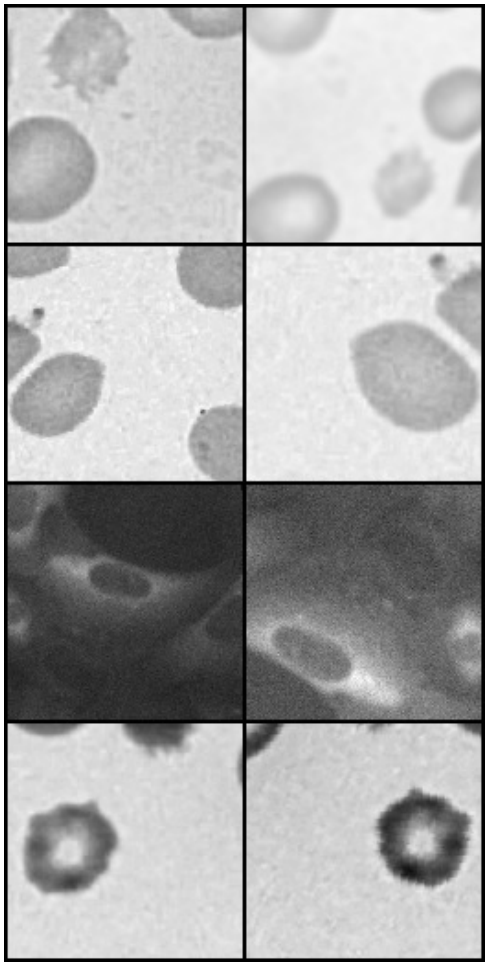

Fig. 5. Farthest duplicate images pairs in the test set, using VGG19 model. Query image is on the left (Column 1), and examples of model selection of their most distant (but duplicate) associated images are on the right (Column 2).

training set. Such networks extract global descriptors (embeddings) from each image in a pair before comparison (via euclidian distance). AUC was used to quantify the performance of the models on the RWE, MFND IND, and BINDER test set. The VGG19 + GeM model gave the best results in detecting same-source image manipulations within the BINDER test set with an AUC of 0.63 for hardest negative sampling and 0.99 for random sampling.

One limitation of the tested models is in the *global* nature of the embeddings. Consider for example images I and J sharing content x and images J and K sharing content y (but not x). The reason why (I,J) is a near-duplicate is not the same as the reason why (J,K) is. That is, there's no global descriptor for J that would make (I,J) and (J,K) near-duplicates without also making (I,K) near-duplicates. Thus, using local feature descriptors may yield improved results.

Potential Impact and Application Considerations

A tool that has the potential of identifying and assessing possibly duplicative images in scientific literature can be extremely useful for our community in the surveillance and maintenance of research data integrity. As we contemplated in the Introduction of this paper, there are various research workflows and pipelines where automated image detection platforms could provide quality assurance check-points that have to-date been largely absent or reliant on individual, manual, "by eyesight" scanning. However, as algorithms and tools (such as the platform our team is developing) become more publicly available, the scientific and research integrity community should consider all ways in which such tools may be utilized by our colleagues. Instances of duplication within the literature could be genuine errors or the authors might have valid-reasons for reusing images. Each scenario of potential image duplication/manipulation is unique, and hence requires thoughtful discussion and review with authors and associated colleagues and stakeholders invested in the recording or reporting of associated research data.

Finally, with respect to benchmark and standards development, we can envision many ways in which our BINDER dataset and our synthetic dataset methodology could prove

useful to other teams. Certainly the goal of the BINDER dataset and our intention behind establishing it was to aid the design and development of models that detect near-duplicate images in biomedical literature. However, we foresee additional applications where either the BINDER data set, the dataset methodology, or both, may assist teams developing other types of image retrieval technologies (e.g., medical image retrieval platforms for improved diagnosis and treatment).

ACKNOWLEDGMENTS

We'd like to thank the following individual(s) and open source environments for source images to build BINDER: Mikel Ariz for the Adiposoft dataset[6]; Marcelo Cicconet for the the Mouse Embryo Tracking Database[7]; Broad Institute for the Broad Bioimage Benchmark Collection (BBBC) [8]; Open Microscopy for the Image Data Resource (IDR)[9].

This work was supported by HHS ORI grants 1 ORIIR180043-01-00 and 1 ORIIR190048-01-00.

REFERENCES

[1] Daniel E Acuna, Paul S Brookes, and Konrad P Kording. Bioscience-scale automated detection of figure element reuse. *bioRxiv*, 2018, DOI: 10.1101/26941.

[2] Elisabeth M Bik, Arturo Casadevall, and Ferric C Fang. The prevalence of inappropriate image duplication in biomedical research publications. *MBio*, 7(3):e00809–16, 2016, DOI: 10.1128/mBio.00809-16.

[3] Elisabeth M Bik, Ferric C Fang, Amy L Kullas, Roger J Davis, and Arturo Casadevall. Analysis and correction of inappropriate image duplication: the molecular and cellular biology experience. *Molecular and Cellular Biology*, 38(20):e00309–18, 2018, DOI: 10.1128/MCB.00309-18.

[4] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997, DOI: 10.1016/S0031-3203(96)00142-2.

[5] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[6] Richard Connor, Stewart MacKenzie-Leigh, Franco Alberto Cardillo, and Robert Moss. Identification of mir-flickr near-duplicate images: a benchmark collection for near-duplicate detection. In *10th International Conference on Computer Vision Theory and Applications (VISAPP 2015)*, pages 565–571, 2015, DOI: 10.5220/0005359705650571.

[7] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017, DOI: 10.1007/s11263-017-1016-8.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016, DOI: 10.1109/CVPR.2016.90.

[9] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 39–43, 2008, DOI: 10.1145/1460096.1460104.

[10] Marek Jakab and Wanda Benesova. Unsupervised similarity learning from compressed representations via siamese autoencoders. In *Eleventh International Conference on Machine Vision (ICMV 2018)*, volume 11041, page 110412F. International Society for Optics and Photonics, 2019, DOI: 10.1117/12.2522920.

[11] Yan Ke, Rahul Sukthankar, and Larry Huston. Efficient near-duplicate detection and sub-image retrieval. In *Acm Multimedia*, volume 4, page 5. Citeseer, 2004, DOI: 10.1145/1027527.1027729.

[6]https://imagej.net/Adiposoft

[7]http://celltracking.bio.nyu.edu/

[8]https://data.broadinstitute.org/bbbc/

[9]https://idr.openmicroscopy.org/

[12] Saehoon Kim, Xin-Jing Wang, Lei Zhang, and Seungjin Choi. Near duplicate image discovery on one billion images. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 943–950. IEEE, 2015, DOI: 10.1109/WACV.2015.130.

[13] Yang-Min Kim, Jean-Baptiste Poline, and Guillaume Dumas. Experimenting with reproducibility: a case study of robustness in bioinformatics. *GigaScience*, 7(7), 06 2018. giy077, DOI: 10.1093/gigascience/giy077.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014, DOI: 10.1007/978-3-319-10602-148.

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004, DOI:10.1023/B:VISI.0000029664.99615.94.

[16] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017, DOI: 10.5555/3295222.3295236.

[17] Lia Morra and Fabrizio Lamberti. Benchmarking unsupervised near-duplicate image detection. *Expert Systems with Applications*, 135:313–326, Nov 2019, DOI: 10.1016/j.eswa.2019.05.002.

[18] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018, DOI: 10.1109/TPAMI.2018.2846566.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Lev V Utkin, Vladimir S Zaborovsky, Alexey A Lukashin, Sergey G Popov, and Anna V Podolskaja. A siamese autoencoder preserving distances for anomaly detection in multi-robot systems. In *2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 39–44. IEEE, 2017, DOI: 10.1109/ICCAIRO.2017.17.

[21] Kirstie Whitaker. Showing your working: a how to guide to reproducible research. 8 2017, DOI: 10.6084/m9.figshare.4244996.v2.

[22] Ziyue Xiang and Daniel Acuna. Scientific image tampering detection based on noise inconsistencies: A method and datasets. *arXiv preprint arXiv:2001.07799*, 2020, DOI: 10.21203/rs.2.22943/v1.