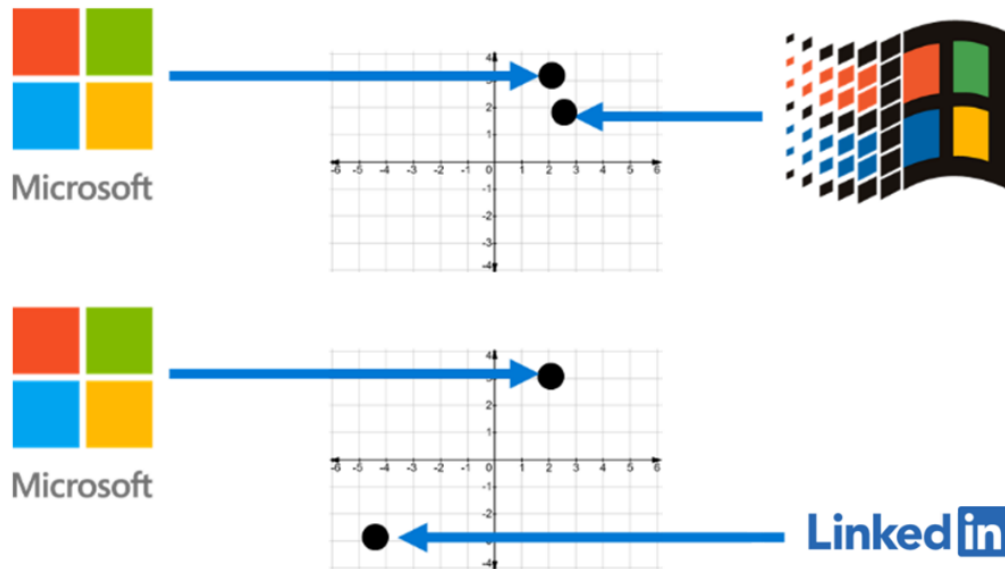


Spotting brand impersonation with Swin transformers and Siamese neural networks

microsoft.com/security/blog/2021/08/04/spotting-brand-impersonation-with-swin-transformers-and-siamese-neural-networks/

August 4, 2021



Every day, [Microsoft Defender for Office 365](#) encounters millions of brand impersonation emails. Our security solutions use multiple detection and prevention techniques to help users avoid divulging sensitive information to phishers as attackers continue refining their impersonation tricks. In this blog, we discuss our latest innovation toward developing another detection layer focusing on the visual components of brand impersonation attacks. We presented this approach in our Black Hat briefing [Siamese neural networks for detecting brand impersonation](#) today.

Before a brand impersonation detection system can be trained to distinguish between legitimate and malicious email that use the same visual elements, we must first teach it to identify what brand the content is portraying in the first place. Using a combination of machine learning techniques that convert images to real numbers and can perform accurate judgments even with smaller datasets, we have developed a detection system that outperforms all visual fingerprint-based benchmarks on all metrics while maintaining a 90% hit rate. Our system is not simply “memorizing” logos but is making decisions based on other salient aspects such as color schemes or fonts. This, among other state-of-the-art AI that feeds into [Microsoft 365 Defender](#), improves our protection capabilities against the long-standing problem of phishing attacks.

Two-step approach to spot impersonations

In brand impersonation attacks, an email or a website is designed to appear visually identical to a known legitimate brand, like Microsoft 365 or LinkedIn, but the domain—to which user-inputted information, like passwords or credit card details, is sent—is actually controlled by an attacker. Examples of a malicious sign-in page impersonating Microsoft is shown in Figure 1.

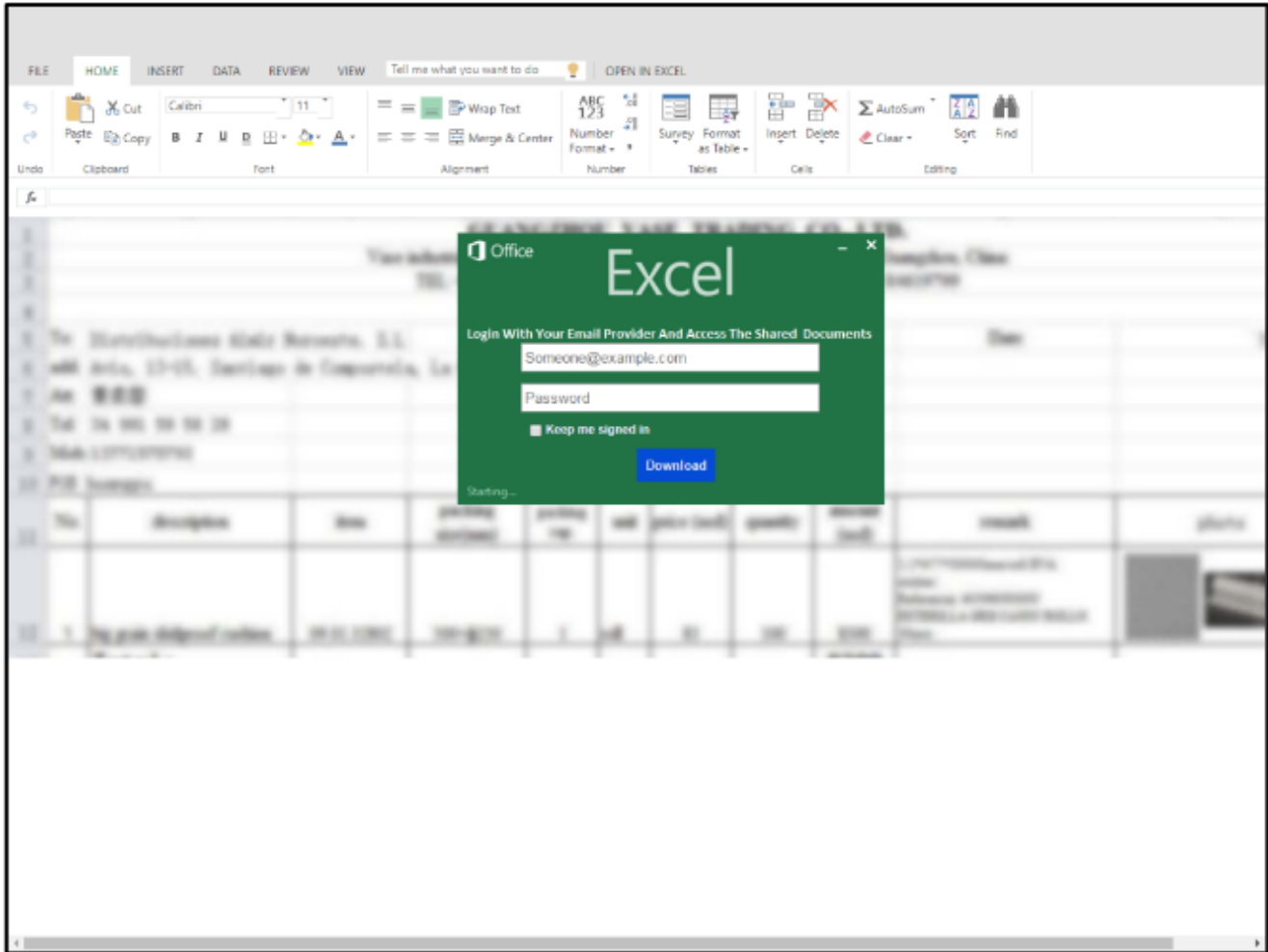


Figure 1. Example of a Microsoft brand impersonation attempt

Any vision-based system, computer or human, that detects brand impersonation attacks must take a two-step approach upon receiving content:

1. Determine whether the content looks like content from a known brand, and if so, which brand
2. Determine if other artifacts associated with the content (such as URLs, domain names, or certificates) match those used by the identified brand

For example, if a brand impersonation detection system sees an image that appears to come from Microsoft but also notices that the URL is indeed from Microsoft and that the certificate matches a known certificate issued to Microsoft, then the content would be classified as legitimate.

However, if the detector encounters content which shares visual characteristics with legitimate Microsoft content like in Figure 1, but then notices that the URL associated with the content is an unknown or unclassified URL with a suspicious certificate, then the content would be flagged as a brand impersonation attack.

Training our system to identify brands

The key to an effective brand impersonation detection system is identifying known brands as reliably as possible. This is true for both a manual system and an automated one. For sighted humans, the process of identifying brands is straightforward. On the other hand, teaching an automated system to identify brands is more challenging. This is especially true because each brand might have several visually distinct sign-in pages.

For example, Figure 2 shows two Microsoft Excel brand impersonation attempts. While both cases share some visual characteristics, the differences in background, color, and text make the creation of rule-based systems to detect brands based on rudimentary similarity metrics (such as robust image hashing) more difficult. Therefore, our goal was to improve brand labeling, which will ultimately improve brand impersonation detection.

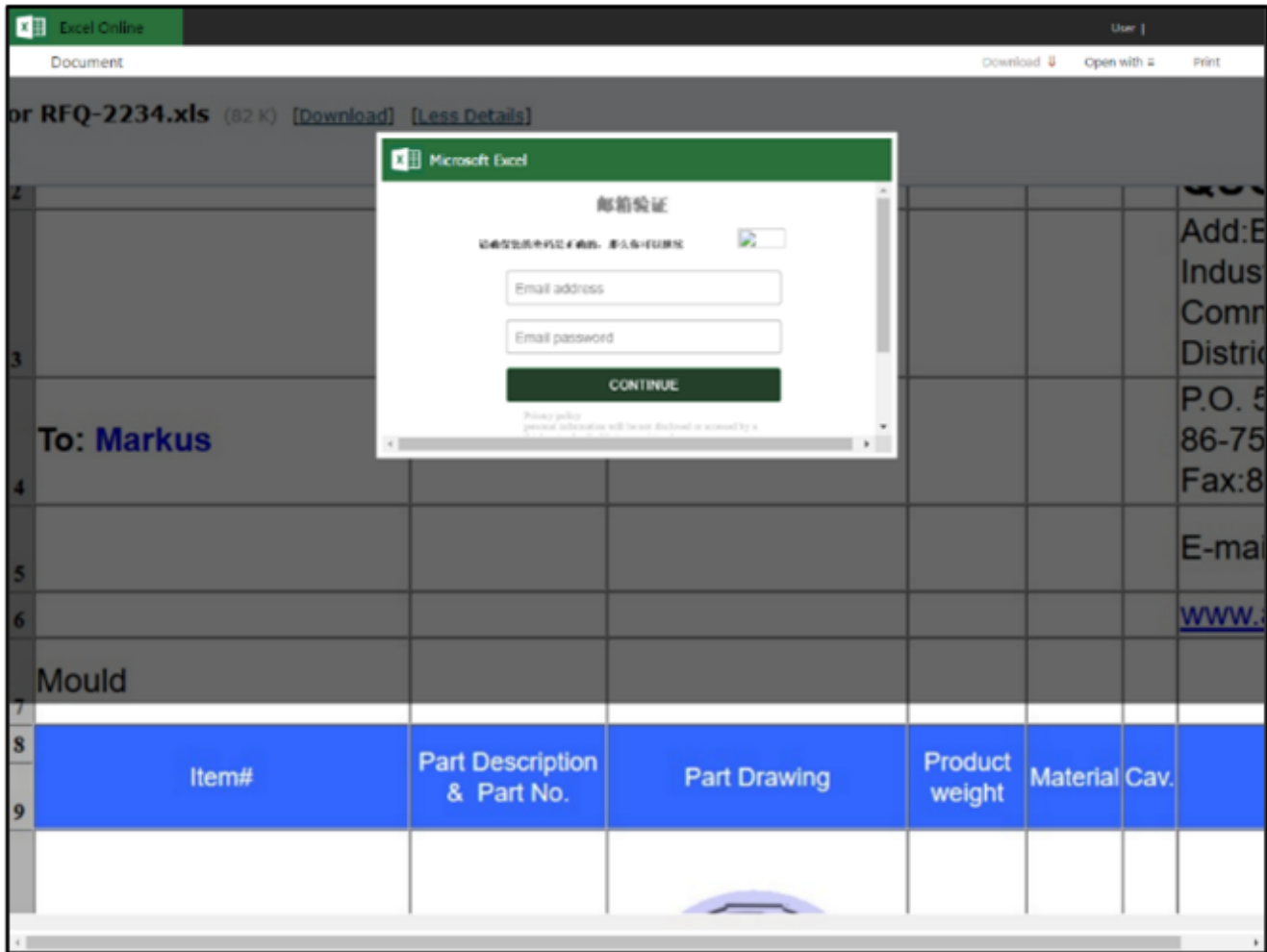


Figure 2. Another examples of brand impersonation attempt targeting Microsoft Excel

Of course, deep learning is the assumed default tool for image recognition, so it was only natural to perform brand detection by combining labeled brand images with modern deep-learning techniques. To do this, we first sought out, captured, and manually labeled over 50,000 brand impersonation screenshots using our own detonation system.

While our dataset consisted of over 1,300 distinct brands, most brands were not well-represented. Appearing less than 5 times are 896 brands while 541 brands only appeared in the dataset once. The lack of significant representation for each brand meant that using standard approaches like a convolutional neural network would not be feasible.

Converting images to real numbers via embeddings

To address the limitations of our data, we adopted a cutting-edge, few-shot learning technique known as Siamese neural networks (sometimes called neural twin networks). However, before explaining what a Siamese neural network is, it is important to understand how embedding-based classifiers work.

Building an embedding-based classifier proceeds in two steps. The first step is to embed the image into a lower dimensional space. All this means is that the classifier transforms the pixels that make up the images into a vector of real numbers. So, for example, the network might take as an input the pixel values in Figure 1 and output the value (1.56, 0.844). Because the network translates the images into *two* real numbers, we say the network embeds the images into a *two-dimensional* space.

While in practice we use more than a two-dimensional embedding, Figure 3 shows all our images embedded in two-dimensional space. The red dots represent the embeddings of images all appearing to be from one brand. This effectively translates the visual data into something our neural network can digest.

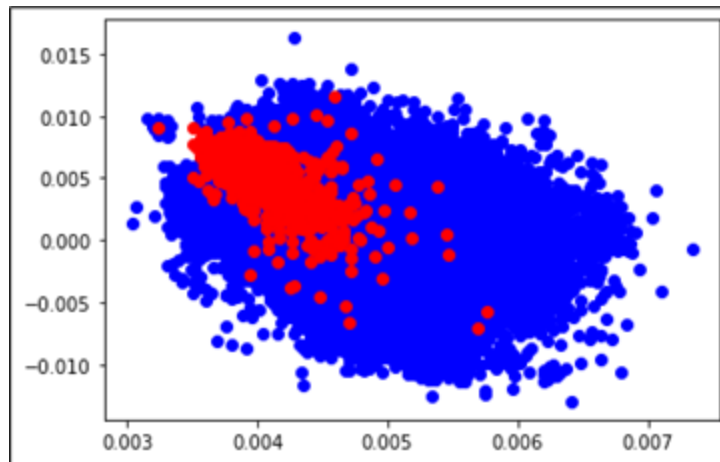


Figure 3: A two-dimensional representation of embeddings, where the red dots represent one brand

Given the embeddings, the second step of the algorithm is to classify the embedded images. For example, given a set of embedded screenshots and a new screenshot we call X , we can perform brand classification by embedding X and then assigning to X the brand whose image is “closest” to X in the embedded space.

Training the system to minimize contrastive loss

In understanding the two-dimensional embeddings above, readers might assume that there was an “embedder” that placed screenshots of the same brand close together, or at least that there was some inherent meaning in the way the images were embedded. Of course, neither was true. Instead, we needed to train our detector to do this.

This is where Siamese neural networks with an associated contrastive loss come into play. A Siamese network takes as an input *two* raw images and embeds them both. The *contrastive loss* the network computes is the distance between the images if the images come from the same brand and the negative of the distance between the images if they come from a

different brand. This means that when a Siamese network is trained to *minimize* losses, it embeds screenshots of the same brand close together and screenshots of different brands far apart. An example of how the network minimizes losses is shown in Figure 4.

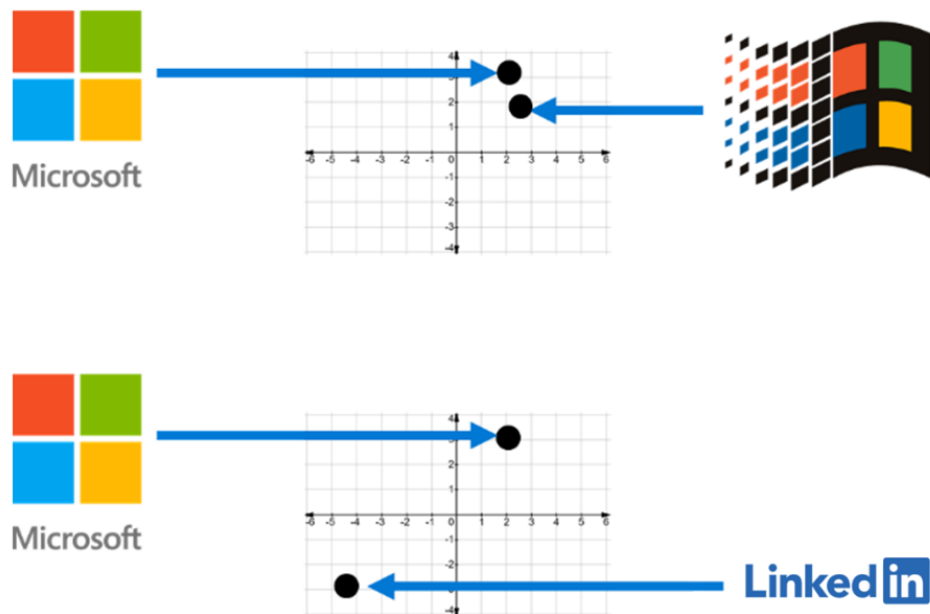


Figure 4. Successful Siamese network embeddings. The network minimizes loss by embedding screenshots that pertain to Microsoft close together while simultaneously embedding screenshots from Microsoft and LinkedIn far apart. Note that the algorithm is trained on entire screenshots and not just logos. The logos are used here for illustrative purposes only.

We also mentioned that the Siamese network can perform any type of classification on the embedded images. Therefore, we used standard feedforward neural networks to train the system to perform the classification. The full architecture is illustrated in Figure 5 below. The images were first embedded into a low dimensional space using Swin transformers, a cutting edge computer-vision architecture. The embeddings were then used to calculate the contrastive loss. Simultaneously, the embeddings were fed into a feedforward neural network which then outputted the predicted class. When training the system, the total loss is the sum of the contrastive loss and a standard log-likelihood loss based on the output of both classification networks.

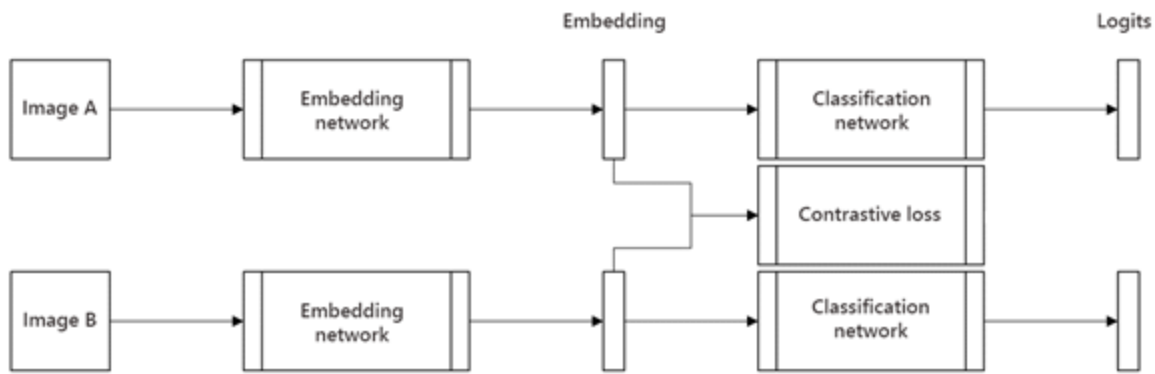


Figure 5. Siamese neural network architecture

Basing success metrics on costs and benefits of correct labelling

Since this is a multi-class classification system, we needed to be careful about how we defined our metrics for success. Specifically, the notions of a true positive or a false negative are not well-defined in multi-class classification problems. Therefore, we developed metrics based on the associated costs and benefits of real-world outcomes. For example, the cost of mislabeling a known brand as another known brand is not the same as observing a never-before-seen brand but labeling it as a known brand. Furthermore, we separated our metrics for known and unknown brands. As a result, we developed the following five metrics:

1. Hit rate – the proportion of known brands that are correctly labeled
2. Known misclassification rate – the proportion of known brands that are incorrectly labeled as another known brand
3. Incorrect unknown rate – the proportion of known brands that are incorrectly labeled as an unknown brand
4. Unknown misclassification rate – the proportion of screenshots of unknown brands that are labeled as a known brand
5. Correct unknown rate – the proportion of unknown brands that are correctly labeled as unknown

These metrics are also summarized in Figure 6 below. Since all our images were labeled, we simulated an unknown brand by removing all brands with only one screenshot from the training set and only used them for evaluating our metrics on a held-out test set.







Metric	Actual	Prediction
▲ Known hit rate	 Microsoft	 Microsoft
▼ Known misclassification rate	 Microsoft	
▼ Incorrect unknown rate	 Microsoft	?
▼ Known misclassification rate	?	 Microsoft
▲ Correct unknown rate	?	?

Figure 6. Classification metrics. Metrics with upward-facing triangles indicate that the results are better when they are higher. Metrics with downward-facing triangles are better when they are lower.

Outperforming visual fingerprint-based benchmarks

The main results of our brand impersonation classification system are given in Figure 7 but are straightforward to summarize: **Our system outperforms all visual fingerprint-based benchmarks on all metrics while still maintaining a 90% hit rate.** The results also show that if instead of maximizing hit rate, it was more beneficial to minimize the known

misclassification rate, it is possible to have the known misclassification rate be less than 2% while the hit rate remains above 60% and the Siamese network still beats the visual fingerprint-based approaches on all metrics.

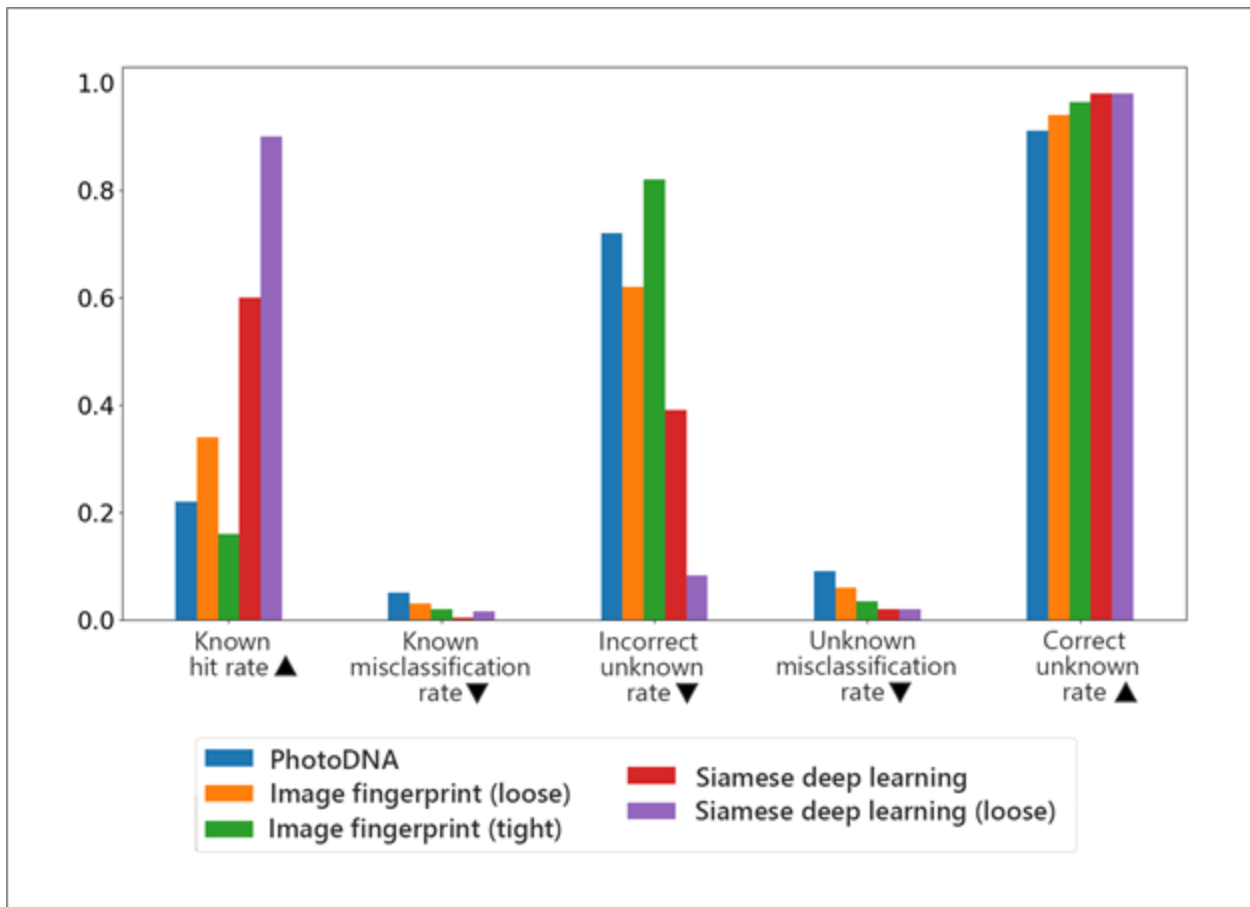


Figure 7. Results of how our system fared against other image recognition systems

We can further examine some examples to show that the network did not simply memorize the screenshots and can correctly label variations on the same brand. Figure 8 shows two different malicious DHL brand impersonation sign-in pages. Despite a different visual layout and color scheme (use of a black bar in the left image, white on the right), the network still correctly classified both. Furthermore, the network was able to correctly classify the image on the left even though it carried several logos of other companies on the bottom bar. This means that the network is doing more than just logo recognition and making decisions based on other features such as color schemes or the dominant font style.

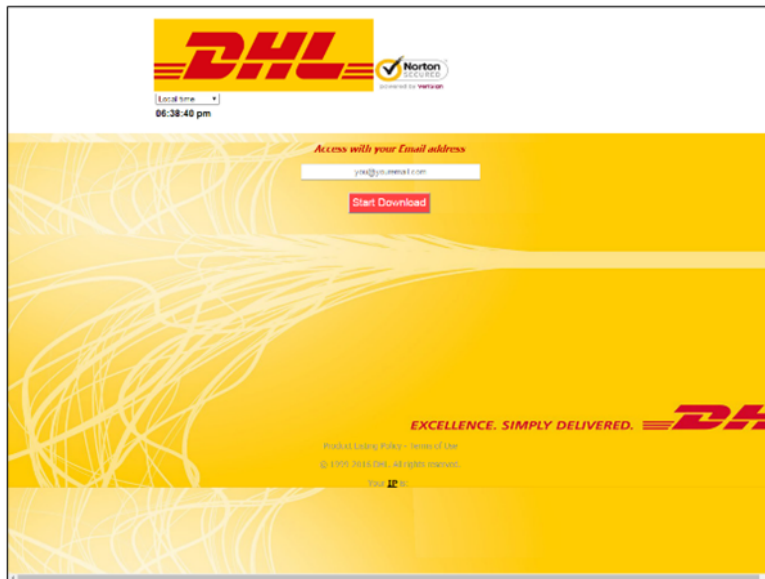
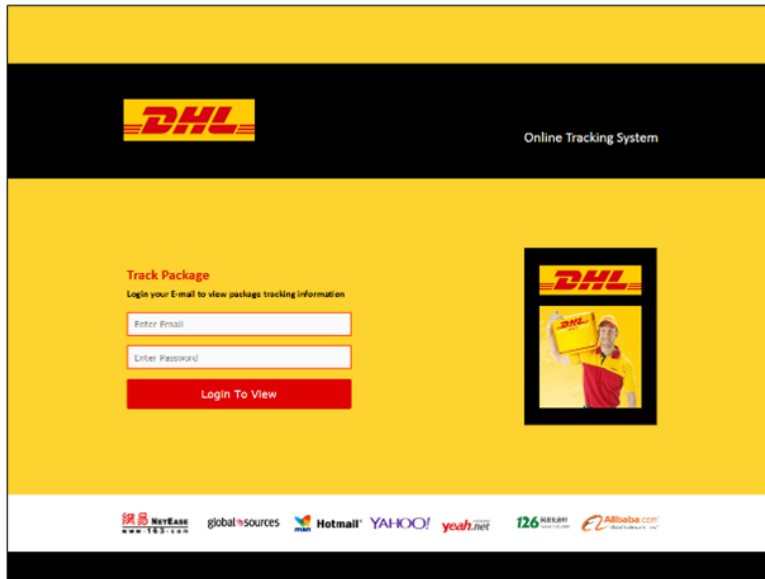


Figure 8. Variations on the DHL sign-in page, both classified correctly by our system as pertaining to DHL

Important applications in detecting phishing campaigns

Phishers have become particularly good at creating phishing websites or crafting emails that closely resemble known legitimate brands visually. This allows them to gain users' trust and trick them into disclosing sensitive information.

Our work prevents attackers from hijacking legitimate brands by detecting entities that visually look like legitimate brands but do not match other known characteristics or features of that brand. Moreover, this work helps us with threat intelligence generation by clustering

known attacks or phishing kits based on the specific brands they target visually and identifying new attack techniques that might impersonate the same brand but employ other attack techniques.

Dedicated research teams in Microsoft stay on top of threats by constantly improving the AI layers that support our threat intelligence which then feeds into our ability to protect against and detect threats. Microsoft Defender for Office 365 protects against email-based threats like phishing and empowers security operations teams to investigate and remediate attacks. Threat data from Defender for Office 365 then increases the quality of signals analyzed by Microsoft 365 Defender, allowing it to provide cross-domain defense against sophisticated attacks.

Justin Grana, Yuchao Dai, Jugal Parikh, and Nitin Kumar Goel

Microsoft 365 Defender Research Team