Design Patent Similarity with ArcFace Metric Learning

Hui-Ching Chuang
National Taipei University
New Taipei City, Taiwan
hcchuang@gm.ntpu.edu.tw

Yueh-Heng Wu
National Taipei University
New Taipei City, Taiwan
ygsdsf111@gmail.com

Abstract

We propose an embedding-based patent retrieval system that converts patent drawings into compact visual representations for scalable similarity search. Using patent-identity supervision with ArcFace metric learning and ConvNeXtV2 backbones on an extensive U.S. design patent corpus (1976–2025), our best 256-pixel ConvNeXtV2-Tiny model reaches mAP = 0.819 on the DeepPatent [15] Standard Gallery and mAP = 0.649 on the substantially larger Extended Gallery. Query expansion further raises Standard-Gallery mAP to 0.842. Experiments show that higher-resolution inputs and retrieval re-ranking improve fine-grained visual matching, while the ArcFace-trained embeddings provide strong instance-level retrieval performance for practical prior-art search and design patent analytics.

CCS Concepts

• **Information systems → Specialized information retrieval**; • **Computing methodologies → Computer vision**.

Keywords

Design Patent Retrieval, ArcFace, Deep Metric Learning, ConvNeXtV2, Graph Diffusion

1 Introduction

Design patent filings have expanded dramatically, motivating the development of scalable, data-driven methods. Annual applications rose from about 7,061 in 1976 to over 65,000 in 2025. Unlike utility patents, design patents are primarily defined by technical drawings, so effective analysis hinges on robust visual representations of the patent figures. In addition, design patents must satisfy the non-obviousness requirement (35 U.S.C. § 103); because the claim is embodied in the figures, examination often amounts to searching a large corpus for visually similar prior-art designs. These considerations motivate embedding-based retrieval systems that learn compact drawing embeddings and support accurate, efficient prior-art search at scale.

Earlier patent-image studies primarily relied on supervised classifiers, training CNNs to predict predefined categories (e.g., IPC codes or visual types) to inject domain knowledge and capture geometric cues [12, 14, 23]. However, because these methods depend on coarse category labels, the resulting features may be less suitable for instance-level similarity search, motivating later work on deep metric learning for patent-image retrieval [10, 15]. Recent surveys likewise document the growing role of multimodal and representation-learning methods across patent-analysis tasks [22].

To avoid reliance on subjective pairwise similarity annotations while still exploiting reliable administrative information, we use each patent ID as a distinct training target and optimize an ArcFace additive angular-margin objective [5]. Images belonging to the same design patent share one patent-identity label, and the angular margin encourages compact within-patent representations while increasing separation from other patent identities. We evaluate this objective function with ResNet and ConvNeXtV2 backbones [9, 16, 24]. ConvNeXtV2 is particularly attractive for sparse technical drawings because it combines a modern convolutional framework with large receptive fields and Global Response Normalization. Related work also uses ConvNeXtV2 to study designers' style shifts [3].

To further refine retrieval precision at scale, we incorporate re-ranking techniques [4, 6], including query expansion and sparse graph diffusion. These post-processing steps exploit neighborhood information in the learned embedding space to reduce ranking errors caused by visually similar but distinct design patents.

We assemble a large-scale U.S. design-patent visual archive spanning 1976–2025, containing 834,168 unique design patents and 5,200,417 image instances. On the DeepPatent benchmark [15], our ArcFace with ConvNeXtV2-Tiny model at 256 pixels achieves mAP = 0.819 on the Standard Gallery and 0.649 on the Extended Gallery; query expansion increases Standard-Gallery mAP to 0.842. These results substantially exceed the original DeepPatent benchmark while avoiding claims that depend on subjective human similarity labels.

Our contributions are threefold: (i) we develop a patent-identity-supervised ArcFace retrieval framework with modern vision backbones; (ii) we deliver an accurate, scalable pipeline for visual similarity search without requiring manually annotated image-pair similarity labels; and (iii) we validate the framework through large-scale benchmark experiments, re-ranking analyses, qualitative retrieval cases, and direct analysis of the learned embedding space.

The remainder of the paper is organized as follows. Section 2 introduces our patent-image retrieval framework, and Section 3 describes dataset construction, pre-processing, and experimental results. Section 4 presents re-ranking and embedding-space analyses. Section 5 concludes.

2 Methodology

This section introduces our visual embedding approach, which integrates ArcFace metric learning with ResNet and ConvNeXtV2 backbones. We also define the performance metrics used in our experiments.

2.1 ArcFace metric learning with ConvNeXtV2 backbone

Figure 1 summarizes the training and retrieval pipeline. We write each training observation as a pair (x_i, y_i). Here, x_i is the i-th design-patent drawing sheet drawn from the training pool described in

Section 3.1, and y_i is the patent ID associated with that drawing. Let N_{pat} denote the number of distinct patent IDs in the training set. For model training, each patent ID is encoded as an integer identity label $y_i \in \{1, \ldots, N_{\text{pat}}\}$. All drawings belonging to the same patent therefore share the same y_i.

Before entering the network, x_i is randomly augmented using the procedure in Section 3.2, producing $\tilde{x}_i$. The backbone $f_\psi(\cdot)$ is the image feature extractor, and ψ denotes its learned parameters. It converts $\tilde{x}_i$ into a feature vector: a compact numerical description of the drawing. We L2-normalize this vector to obtain the embedding used for retrieval,

$$\mathbf{q}_i = \frac{f_\psi(\tilde{x}_i)}{\|f_\psi(\tilde{x}_i)\|_2}. \tag{1}$$

Thus, $\|\mathbf{q}_i\|_2 = 1$. Normalization removes vector magnitude, so similarity depends only on the direction of $\mathbf{q}_i$. ArcFace uses the patent label y_i directly and does not require explicit training pairs or triplets [5, 10].

For each training patent ID j, the ArcFace head learns a reference vector $\mathbf{w}_j$ that represents patent identity j during training. This vector is a learned model parameter, not an input drawing or the average of the patent's drawings. We normalize it as $\widehat{\mathbf{w}}_j = \mathbf{w}_j / \|\mathbf{w}_j\|_2$. The similarity between image i and the reference vector for patent ID j is

$$\cos\theta_{ij} = \widehat{\mathbf{w}}_j^\top \mathbf{q}_i. \tag{2}$$

where θ_{ij} is the angle between $\mathbf{q}_i$ and $\widehat{\mathbf{w}}_j$. A larger cosine means a smaller angle and therefore greater similarity. Because y_i is the patent ID label of image i, $\widehat{\mathbf{w}}_{y_i}$ is the reference vector for the correct patent.

ArcFace introduces two positive scalar hyperparameters that must be chosen before training. The angular margin m controls how much additional separation the model must create between the correct patent and competing patents. The scale s enlarges the bounded cosine scores before softmax so that cross-entropy produces useful gradients. Neither scalar is learned. We choose $m = 0.50$ radians and $s = 30$ and keep them fixed throughout training.

For the correct patent, canonical ArcFace replaces $\cos\theta_{iy_i}$ with $\cos(\theta_{iy_i} + m)$ [5]. Adding m lowers the correct-prediction score and therefore makes the training requirement more demanding. To keep this score decreasing monotonically when the target angle is extremely large, our implementation uses the following safeguarded transformation:

$$\phi_{iy_i} = \begin{cases} \cos(\theta_{iy_i} + m), & \cos\theta_{iy_i} > \cos(\pi - m), \\ \cos\theta_{iy_i} - \sin(\pi - m)m, & \text{otherwise.} \end{cases} \tag{3}$$

The first branch is the canonical ArcFace transformation. The second branch is used only for an extremely large target angle and keeps the transformed score monotonic. A *logit* is the unnormalized score for one patent ID supplied to softmax. Only the correct-patent logit receives the margin; all competing patents retain their original cosine scores. After multiplication by s, the logits are

$$g_{ij} = \begin{cases} s\phi_{iy_i}, & j = y_i, \\ s\cos(\theta_{ij}), & j \neq y_i, \end{cases} \tag{4}$$

Let B be the number of images processed in one physical mini-batch. Cross-entropy converts the logits into probabilities over the training patent IDs and penalizes the model when the correct patent does not receive a sufficiently high probability:

$$\mathcal{L}_{\text{Arc}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(g_{iy_i})}{\sum_{j=1}^{N_{\text{pat}}} \exp(g_{ij})}. \tag{5}$$

Minimizing this loss increases the correct-patent logit relative to the $N_{\text{pat}} - 1$ competing logits. Because the margin lowers the correct-patent score, the backbone must place $\mathbf{q}_i$ closer to the reference vector for its correct patent than ordinary normalized softmax would require. Images from the same patent consequently become close because they are trained toward the same reference vector, even though the loss does not directly compare image pairs.

The patent-ID reference vectors and ArcFace training head are needed only during training. At retrieval time, we discard the head, compute the normalized image embeddings in Eq. (1), and rank gallery images by cosine similarity.

We evaluated four backbone architectures: ResNet-18, ResNet-50, ResNet-152, and ConvNeXtV2-Tiny. The ResNet family provides conventional convolutional baselines [9], while ConvNeXtV2-Tiny provides a modern ConvNet comparison [24].

ConvNeXtV2 [24] is a modern ConvNet that utilizes large (7×7) depthwise kernels and a Global Response Normalization (GRN) layer to enhance feature diversity. We implemented the model using the PyTorch Image Models (`timm`) library[1], specifically adopting the `convnextv2_tiny.fcmae_ft_in22k_in1k` variant. This model is initialized from weights obtained through the ConvNeXtV2 pretraining/fine-tuning recipe.[2] In terms of capacity, ConvNeXtV2-Tiny contains about 28.6M parameters, placing it between ResNet-18 and ResNet-152 and close to ResNet-50 in scale.

2.2 Retrieval evaluation metrics

To evaluate retrieval performance, we report a unified set of ranking and coverage metrics [15, 18]. Let Q denote the set of query images and let N_{db} denote the number of gallery images. Each query image is one evaluation unit. The official query and gallery lists are disjoint, so a query cannot retrieve itself; every gallery image with the same patent ID as the query is relevant. Every query has at least one relevant gallery image, so $R_q \geq 1$, and we consider cutoffs $1 \leq K \leq N_{\text{db}}$. For each query $q \in Q$, let $\text{rel}_q(r) \in \{0, 1\}$ indicate whether the gallery item at rank r is relevant. The total number of relevant gallery items for query q is

$$R_q = \sum_{r=1}^{N_{\text{db}}} \text{rel}_q(r),$$

and precision at rank r is

$$P_q(r) = \frac{1}{r}\sum_{t=1}^{r} \text{rel}_q(t).$$

The Average Precision for query q is

$$AP(q) = \frac{1}{R_q}\sum_{r=1}^{N_{\text{db}}} P_q(r)\,\text{rel}_q(r),$$

[1] https://github.com/huggingface/pytorch-image-models
[2] https://github.com/facebookresearch/ConvNeXt-V2

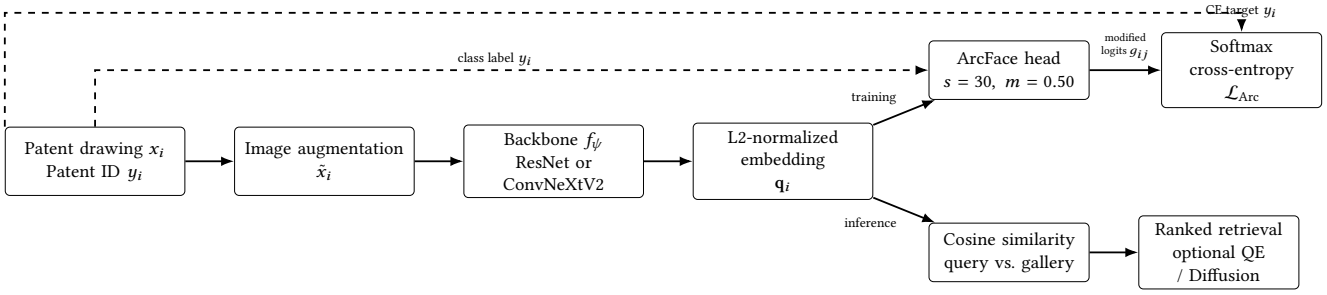


Figure 1: Overview of the ArcFace-based design-patent retrieval framework. During training, each patent ID is a distinct identity label. The ArcFace head applies an additive angular margin to the correct-ID score and produces scaled, margin-modified logits g_{ij}, which are optimized by softmax cross-entropy $\mathcal{L}_{\text{Arc}}$ using y_i as the target. During retrieval, the ArcFace head is discarded; L2-normalized backbone embeddings are compared by cosine similarity and can optionally be re-ranked by query expansion or graph diffusion.

and the Mean Average Precision (mAP) is

$$mAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q). \tag{6}$$

We also report Top-K retrieval accuracy (Acc@K), i.e., the query-level hit rate, which measures whether at least one relevant item appears within the top K results:

$$\text{Acc@}K = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}\left\{ \sum_{r=1}^{K} \text{rel}_q(r) > 0 \right\}.$$

Recall at cutoff K is

$$\text{Rec@}K(q) = \frac{\sum_{r=1}^{K} \text{rel}_q(r)}{R_q}.$$

To capture the rank of the first correct result, let rank_q denote the position of the first relevant item for query q; the Mean Reciprocal Rank (MRR) is

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q}. \tag{7}$$

Because the complete gallery is ranked and $R_q \geq 1$, rank_q exists for every benchmark query; under a truncated evaluation with no relevant result, the reciprocal rank is conventionally set to zero. We further use R-Precision, defined for query q as the precision at rank R_q, $\text{R-Prec}(q) = P_q(R_q)$. Finally, we report normalized discounted cumulative gain (nDCG). With binary relevance, the discounted cumulative gain for query q at cutoff K is

$$DCG_q@K = \sum_{r=1}^{K} \frac{2^{\text{rel}_q(r)} - 1}{\log_2(r+1)}. \tag{8}$$

The corresponding ideal discounted cumulative gain is

$$IDCG_q@K = \sum_{r=1}^{\min(K,R_q)} \frac{1}{\log_2(r+1)},$$

and

$$nDCG_q@K = \frac{DCG_q@K}{IDCG_q@K}.$$

The reported R-Precision, Recall@K, and nDCG@K values are macro-averaged over query images, so every query image contributes equally.

3 Experimental Results

This section describes our data collection pipeline, preprocessing, and sampling scheme for constructing a large U.S. design patent corpus (1976–2025). We also include major experimental results and comparisons.

3.1 Data preprocessing and sample splitting

We collect design-patent files from the USPTO Open Data Portal (ODP) and augment them with filing-year and USPC classification metadata from PatentsView. We treat each drawing sheet as one image instance, even when it contains multiple views (e.g., front, back, or side). For patents granted through 2002, ODP provides only full-document PDFs. We therefore render the pages at 300 DPI using PyMuPDF, apply OpenCV connected-component analysis and geometric cropping to remove bibliographic text, headers, and scan borders, and use PIL ImageChops bounding-box detection to trim excess white space. We center each drawing on a standardized canvas with a 100-pixel white margin and save it as a lossless PNG. For patents granted after 2002, we use the PNG drawing sheets provided directly by ODP. The resulting 1976–2025 corpus contains 834,168 unique patents and 5,200,417 image instances, substantially exceeding previously documented design-patent corpora [1, 15, 21].

For training and validation, we exclude 25,448 patents containing photographic images rather than line drawings.[3] We identify these patents using an entropy-based filter with human validation; details are provided in the Appendix. We also exclude 46,179 patents in the Kucer et al. [15] benchmark to keep it fully external to model training and validation. The two exclusion sets share 1,753 patents, so we remove 69,874 unique patents in total, leaving 764,294 of the original 834,168 patents.

To improve scalability while preserving coverage, we draw a 20% stratified sample by USPC main class and filing year, yielding 152,858 patents. Requiring at least two images per patent leaves 151,634 patents. We split these patents by identity into 128,888 training patents (85%) and 22,746 validation patents (15%), ensuring

[3] Photographs are generally not standard in design patent applications. The USPTO permits black-and-white photographs instead of drawings only as a limited exception (37 CFR § 1.84(b)(1)); in design applications, such photographs must be limited to the claimed design and must not disclose environmental structure (37 CFR § 1.152).

that images from the same patent do not appear in both sets. We stratify by filing year rather than grant year because it better reflects the timing of design creation and the relevant prior-art context.

Table 1 contrasts our corpus with the DeepPatent benchmark [15]. Panel A summarizes the visual corpus coverage, including the patents used for ArcFace training and validation. Panel B reports the merged DeepPatent image-level split sizes used by our evaluation. The Extended Gallery combines the Standard Gallery with the DeepPatent training and validation images, yielding 338,436 gallery images.

Table 1: Dataset scale and evaluation setup.

Panel A: Corpus scale		
	Ours	Kucer et al. [15]
Design Patents	834,168	46,179
Image Instances	5,200,417	351,569
Sampled Patents ($\geq$ 2 images)	151,634	–
Training Patents	128,888	33,364
Validation Patents	22,746	5,888
Time Span	1976–2025	2018–mid-2019
Panel B: Evaluation set		
Dataset Split	Patents	Images
Query Set	6,927	13,133
Standard Gallery	6,927	38,834
Extended Gallery	46,179	338,436

3.2 Experimental setup and baseline comparison

We conducted experiments across 12 configurations (4 backbone architectures: ResNet-18, -50, -152, and ConvNeXtV2-Tiny, $\times$ 3 input resolutions: 64, 128, and 256 pixels). All evaluated backbones were initialized using the `timm` library pretrained weights. For validation, we randomly sample two distinct images from each validation patent, encode the two views. For each batch, we construct the cosine-similarity matrix and measure whether each image matches an image from the same patent; this in-batch Top-1 retrieval accuracy is used for checkpoint selection.

We apply the same augmentation pipeline to all 12 architectures. Random Resized Crop (scale 0.2 $\sim$ 1.0), Random Horizontal/Vertical Flip ($p = 0.5$), Random Rotation (uniformly sampled from $[-15°, 15°]$, applied with $p = 0.5$), Color Jitter (brightness/contrast/saturation = 0.4, hue = 0.1, $p = 0.8$), and Random Grayscale ($p = 0.2$). During evaluation, images are deterministically resized to the target resolution and normalization without stochastic augmentation.

All experiments were run on a workstation with an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM) and 128 GB system RAM. We train each model for 30 epochs using mixed precision [19] and AdamW [17] with a weight decay of 10^{-2}. The physical mini-batch size varies by architecture and resolution to fit GPU memory, while gradient accumulation maintains an effective batch size of approximately 512. Following linear learning-rate scaling, we set the peak learning rate to $10^{-4}(512/256) = 2 \times 10^{-4}$, with a three-epoch linear warmup followed by cosine decay. We use an ArcFace scale of $s = 30$, a margin of $m = 0.50$, and gradient-norm clipping at 1.0. We select the checkpoint with the highest validation in-batch Top-1 retrieval accuracy.

Table 2 compares our ConvNeXtV2-Tiny cosine-retrieval baseline with alternative model and resolution settings. On the Standard Gallery, PatentNet achieves an mAP of 0.376 and Acc@1/5/20 of 0.691/0.784/0.841. Our 256-pixel ConvNeXtV2-Tiny model improves these results to an mAP of 0.819 and Acc@1/5/20 of 0.931/0.979/0.991. On the more challenging Extended Gallery, it achieves an mAP of 0.649 and Acc@1 of 0.809. In comparison, the 256-pixel ResNet-50 baseline obtains mAP values of 0.386 and 0.246 on the Standard and Extended Galleries, respectively.

Table 2 highlights three patterns:

(1) **Architecture**: ConvNeXtV2-Tiny consistently outperforms the ResNet variants. At 256 pixels, Standard-Gallery mAP is 0.819 for ConvNeXtV2-Tiny versus 0.462/0.386/0.342 for ResNet-152/50/18; on Extended, the corresponding mAP values are 0.649 versus 0.301/0.246/0.223.
(2) **Resolution**: Retrieval accuracy improves sharply with input resolution. For ConvNeXtV2-Tiny, mAP rises from 0.545 to 0.819 on Standard (64 to 256 pixels) and from 0.387 to 0.649 on Extended.
(3) **Scale robustness**: Despite the much larger Extended Gallery (338,436 images), ConvNeXtV2-Tiny at 256 pixels maintains mAP = 0.649, Acc@1 = 0.809, and Acc@20 = 0.971.

Figure 2 presents qualitative baseline retrieval examples on the Extended Gallery using the 256-pixel ConvNeXtV2-Tiny ArcFace model. Cases 1–2 retrieve only same-patent drawings among the top five. Case 3 retrieves the correct patent at rank 1 but then encounters highly similar distractors from neighboring patent identities. Case 4 is a hard-negative failure in which the top-ranked results are visually similar but belong to different patents. These examples illustrate both viewpoint robustness and the remaining difficulty of instance-level discrimination among near-duplicate technical drawings.

Figure 3 presents two queries, each paired with its highest- and lowest-similarity gallery items. Panel (a) shows a successful same-patent retrieval, whereas panel (b) shows a hard negative: a visually near-identical design from a different patent receives the highest cosine similarity. Thus, raw cosine similarity captures geometric resemblance but does not always distinguish patent identity.

4 Manifold-based re-ranking and embedding-space analysis

4.1 Manifold-based re-ranking

Standard retrieval ranks gallery items by cosine similarity to the query, treating each query–gallery pair independently. However, cosine similarity can be sensitive to noise and spurious matches in large galleries. To exploit the neighborhood consistency encoded in the embedding manifold, we adopt two re-ranking strategies: query expansion and diffusion [4, 6], which have been validated in various image-retrieval tasks [2, 7, 11].

Query Expansion (QE): To reduce sensitivity to an idiosyncratic query view, we select the top-k nearest neighbors from the

Table 2: Baseline retrieval performance on the Standard and Extended galleries.

Backbone	Input size (px)	Standard Gallery (38,834 images)				Extended Gallery (338,436 images)			
		mAP	Acc@1	Acc@5	Acc@20	mAP	Acc@1	Acc@5	Acc@20
PatentNet (Kucer et al.) [15]	–	0.3760	0.6910	0.7840	0.8410	0.2620	0.5510	–	–
ResNet-18	64	0.1776	0.4276	0.5447	0.6500	0.1103	0.3094	0.4142	0.4995
	128	0.2509	0.5276	0.6506	0.7473	0.1553	0.3924	0.5112	0.6019
	256	0.3422	0.6354	0.7450	0.8274	0.2225	0.4927	0.6200	0.7062
ResNet-50	64	0.2156	0.4856	0.6088	0.7117	0.1324	0.3553	0.4646	0.5570
	128	0.2961	0.5704	0.7050	0.7986	0.1813	0.4288	0.5523	0.6516
	256	0.3856	0.6629	0.7825	0.8565	0.2461	0.5134	0.6457	0.7396
ResNet-152	64	0.2605	0.5306	0.6634	0.7653	0.1566	0.3888	0.5108	0.6159
	128	0.3590	0.6280	0.7576	0.8455	0.2187	0.4636	0.6082	0.7075
	256	0.4619	0.7244	0.8311	0.8951	0.3005	0.5657	0.7054	0.7936
ConvNeXtV2-Tiny	64	0.5447	0.8096	0.8916	0.9319	0.3874	0.6666	0.8041	0.8675
	128	0.7254	0.8959	0.9543	0.9762	0.5477	0.7596	0.8936	0.9396
	256	**0.8193**	**0.9312**	**0.9791**	**0.9906**	**0.6493**	**0.8088**	**0.9290**	**0.9711**

initial cosine-similarity ranking and refine the query by aggregation:

$$\mathbf{q}_{\text{new}} = \text{normalize}\left(\mathbf{q}_{\text{old}} + \frac{\alpha}{k}\sum_{i=1}^{k}\mathbf{q}_i\right). \tag{9}$$

Here, $\mathbf{q}_i$ is the normalized embedding of the i-th neighbor and normalize$(\cdot)$ rescales the vector to unit length.

Sparse Graph Diffusion (Diffusion): To enforce broader neighborhood consistency, we construct a sparse kNN graph over the union of the query set (N_q) and gallery (N_{db}), giving $N = N_q + N_{\text{db}}$ nodes. Let $\mathbf{q}_i$ denote the normalized embedding of node i. We define the nonnegative sharpened affinity

$$w_{ij} = \left[\max\left(0, \mathbf{q}_i^\top \mathbf{q}_j\right)\right]^\rho, \qquad \rho = 7, \tag{10}$$

retain each node itself together with its top-k nearest neighbors (the implementation therefore stores $k+1$ entries per row), and row-normalize the sparse affinity matrix as

$$S = D^{-1}W, \qquad D_{ii} = \sum_j w_{ij}. \tag{11}$$

For the quantitative evaluation, the initial state $F_0 \in \mathbb{R}^{N \times N_q}$ contains the cosine similarities between all nodes and each query, and the propagated scores are updated by

$$F_{t+1} = \gamma S F_t + (1-\gamma)F_0, \tag{12}$$

with $\gamma = 0.5$ for 10 iterations. We sort the gallery rows of the final score matrix to obtain the re-ranked results. We evaluate diffusion only on the Standard Gallery because constructing and propagating the much larger Extended-Gallery graph is computationally expensive.

Table 3 reports re-ranking results for the best 256-pixel ConvNeXtV2-Tiny ArcFace model. The strongest mAP is obtained by QE with $k = 3$ and $\alpha = 1.0$, which increases mAP from 0.8193 to 0.8417. Diffusion also improves mAP over the raw-cosine baseline, with its best setting here reaching 0.8332 at $k = 10$, while retaining very high MRR and Top-k accuracy. In this experiment, QE provides the largest improvement in mAP, while diffusion improves mAP and several ranking/recall metrics over the no-re-ranking baseline.

Figure 4 presents two qualitative examples using the best-performing query expansion setting ($k = 3$, $\alpha = 1.0$). Aggregating the top-k nearest neighbors reinforces consistent patent-identity features, promotes additional same-patent views into the top-5 results, and suppresses visually similar distractors.

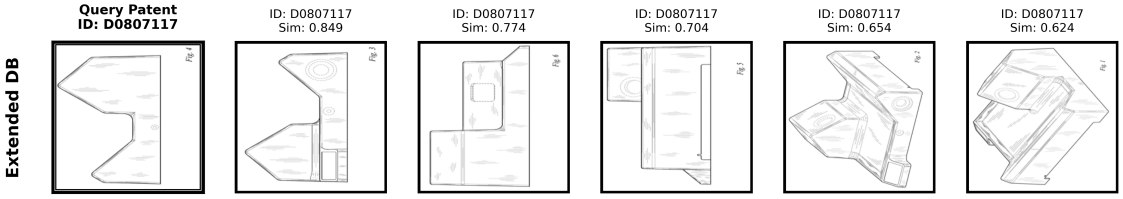
4.2 Embedding-space analysis

Finally, to understand what the ArcFace representation captures beyond aggregate retrieval accuracy, we analyze the cosine geometry of the held-out validation embeddings. We test whether views of the same patent cluster together, whether patent identities remain separable within a shared USPC category, and whether this separation persists against large-pool hard negatives.

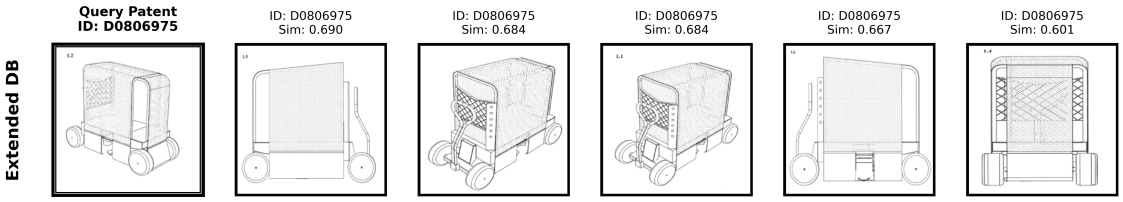
Figure 5a compares 5,000 image pairs in each of four groups: same patent, same subclass but different patents, same main class but different subclasses, and different main classes. Same-patent pairs are drawn from up to 1,800 validation patents, with each patent contributing at most three pairs. Their median cosine similarity is 0.617, compared with 0.013, 0.004, and approximately 0 for the three between-patent groups. Thus, the embeddings primarily capture patent-specific visual form rather than the broader functional categories represented by the USPC hierarchy.

Figure 5b tests whether patent identity remains distinguishable within a single USPC subclass. From D14/299 validation patents with at least eight images, we randomly sample 30 candidates, compute each patent's mean within-patent similarity over $\binom{8}{2} = 28$ image pairs, and select the six closest to the candidate median. Using eight views per patent yields a 48×48 cosine-similarity matrix. The bright 8×8 diagonal blocks and darker off-diagonal regions show that same-patent views remain tightly clustered even when all patents share the same subclass.

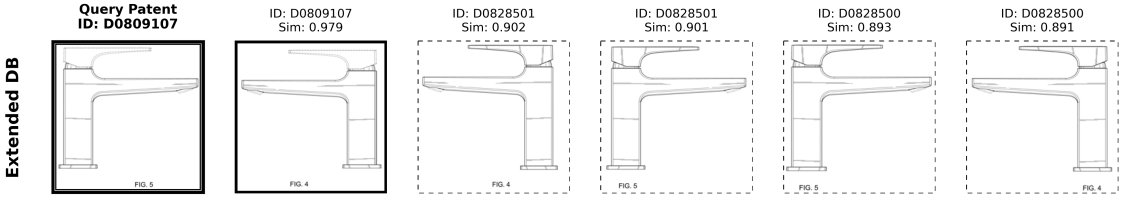
Finally, Figure 5c tests identity separation against a large hard-negative pool. We sample 500 validation patents with at least five views and use exactly five views per patent. The x-axis reports within-patent compactness, measured as the mean cosine similarity across the $\binom{5}{2} = 10$ view pairs. The y-axis reports the similarity between each patent centroid and its most similar different-patent



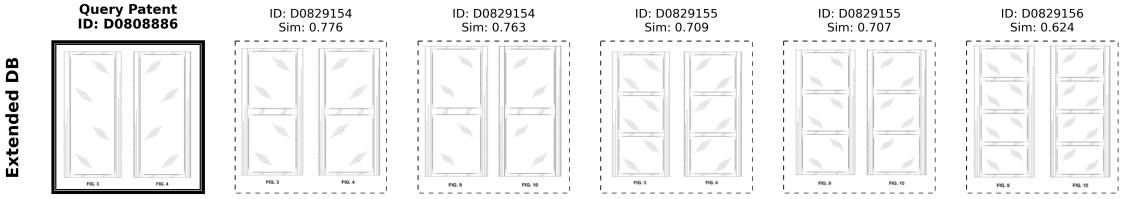
(a) Case 1: Successful retrieval for Patent D0807117.



(b) Case 2: Successful retrieval for Patent D0806975.



(c) Case 3: Correct rank-1 retrieval followed by visually close hard negatives for Patent D0809107.



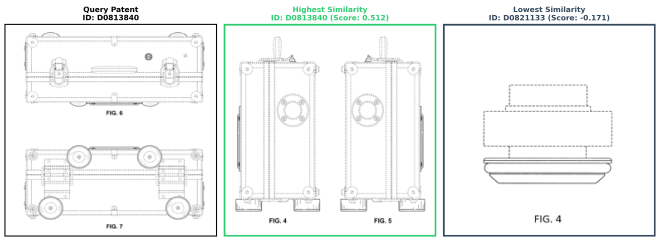
(d) Case 4: Hard-negative failure for Patent D0808886.

Figure 2: Top-5 retrieval results on the Extended Gallery (338,436 images). The query is shown on the left. Solid boxes indicate correct matches (same patent ID), while dashed boxes indicate incorrect distractors (different patent IDs). Panels (a)–(b) show successful retrieval; panel (c) shows a correct rank-1 match followed by visually similar hard negatives, while panel (d) is an incorrect but visually similar retrieval case. Similarity values are cosine similarities between L2-normalized embeddings.

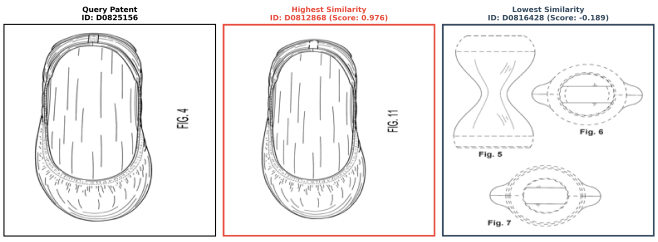
centroid among up to 100,000 candidate patents from the broader corpus, each represented by five views. Points below the dashed line have positive identity margins, whereas points near or above it indicate difficult near-duplicate designs. Colors identify whether the hardest negative shares the same subclass, only the same main class, or neither. Most points below the reference line indicate that patent identity remains distinguishable even after searching up to 100,000 candidate patents, while points near or above the line identify visually near-duplicate designs that remain difficult for the model.

5 Conclusion

We present an end-to-end framework for large-scale design-patent retrieval using ArcFace metric learning and modern vision backbones. Our best configuration, ConvNeXtV2-Tiny at 256 pixels, achieves mAP = 0.819 on the Standard Gallery and 0.649 on the Extended Gallery, while query expansion raises Standard-Gallery mAP to 0.842. The results show substantial gains from stronger backbones, higher input resolution, and retrieval re-ranking. Cosine-space analyses further indicate that the embeddings capture patent-specific visual forms such as shape, geometry, and configuration rather than merely reproducing broad USPC functional categories, although near-duplicate designs remain challenging. As a separate



(a) Successful extreme case: Patent D0813840 retrieves the same patent as its highest-similarity match.



(b) Failure extreme case: Patent D0825156 retrieves the nearly identical Patent D0812868 as its highest-similarity match.

Figure 3: Extreme baseline cosine-similarity examples on the Standard Gallery. Each panel shows a query, its highest-similarity gallery item, and its lowest-similarity item. Panel (a) is a same-patent success; panel (b) is a high-confidence hard negative despite very high visual similarity.

Table 3: Re-ranking results for the 256-pixel ConvNeXtV2-Tiny ArcFace model on the Standard Gallery. Bold denotes the best value in each column (ties included).

Method	Parameters	mAP	MRR	R-Prec	nDCG@20	nDCG@50	Rec@10	Rec@20	Acc@1	Acc@5	Acc@20
Baseline	–	0.8193	0.9525	0.7749	0.8758	0.8892	0.8311	0.8878	0.9312	0.9791	**0.9906**
QE	$k = 10,\ \alpha = 1.0$	0.8125	0.9421	0.7610	0.8714	0.8850	0.8295	0.8946	0.9209	0.9689	0.9869
	$k = 10,\ \alpha = 0.5$	0.8198	0.9478	0.7720	0.8763	0.8898	0.8319	0.8941	0.9267	0.9743	0.9885
	$k = 5,\ \alpha = 1.0$	0.8331	0.9451	0.7866	0.8839	0.8962	0.8466	0.9030	0.9249	0.9742	0.9860
	$k = 5,\ \alpha = 0.5$	0.8327	0.9492	0.7870	0.8843	0.8968	0.8449	0.9006	0.9290	0.9765	0.9877
	$k = 3,\ \alpha = 1.0$	**0.8417**	0.9479	**0.7993**	**0.8892**	**0.9011**	**0.8522**	**0.9052**	0.9285	0.9742	0.9851
	$k = 3,\ \alpha = 0.5$	0.8384	0.9503	0.7948	0.8879	0.9000	0.8493	0.9031	0.9309	0.9753	0.9873
Diffusion	$k = 10,\ \gamma = 0.5$	0.8332	**0.9526**	0.7920	0.8847	0.8973	0.8427	0.8971	**0.9317**	**0.9792**	0.9903
	$k = 5,\ \gamma = 0.5$	0.8321	0.9524	0.7907	0.8837	0.8965	0.8420	0.8959	0.9315	**0.9792**	0.9901
	$k = 3,\ \gamma = 0.5$	0.8301	0.9524	0.7881	0.8825	0.8954	0.8400	0.8946	0.9314	0.9791	0.9900

data contribution, we develop an entropy-based procedure to detect photographic-style patent records and reduce domain mismatch with technical line drawings. Overall, the framework scales to large patent-drawing galleries and supports practical prior-art search and design analytics.

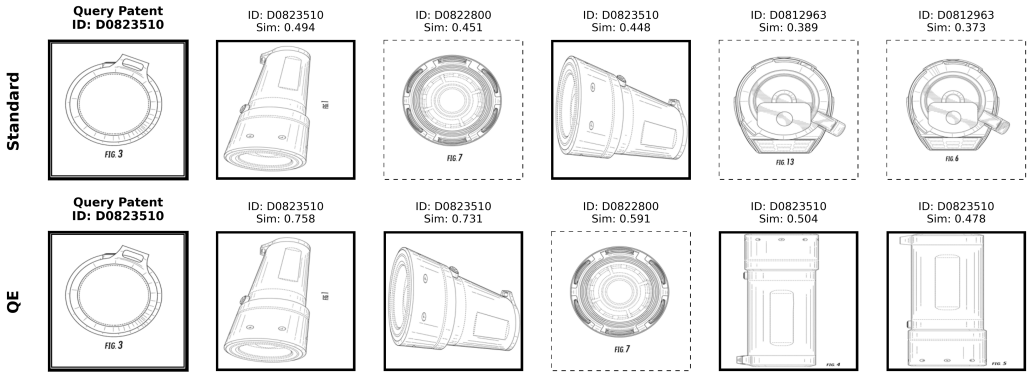
Appendix: Photograph Filtering

Photographic images differ substantially from line drawings in texture and shading, and this discrepancy can hinder learning geometry-focused features. Figure 6a presents examples of photographic-style inputs, characterized by continuous tonal variations and complex textures. To maintain dataset consistency, we filter these images using an entropy-based criterion. This strategy leverages the statistical principle that line drawings possess sparse, bimodal intensity distributions (low entropy), whereas photographic images exhibit complex textural information (high entropy). Based on Shannon's information theory [20], utilizing histogram entropy to characterize image complexity and texture is a well-established practice in computer vision [8, 13]. We define the pixel-intensity entropy H for an image as

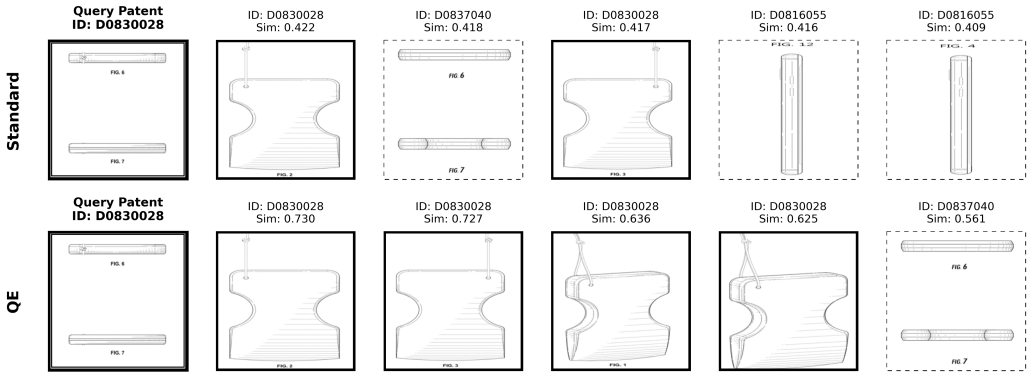
$$H = -\sum_{i=0}^{255} p_i \log_2(p_i).$$

where p_i is the probability of pixel intensity i. Typically, line drawings exhibit low entropy due to sparse information, whereas photographs exhibit high entropy.

We manually labeled 2,504 photographs and compared their entropy distribution with that of a random sample of 25,000 line-drawing patents. Figure 6b shows a clear separation between these two distributions. We set the cutoff at the 1st percentile of the photograph entropy distribution ($\tau = 0.724$), which targets a 99% recall rate for photographs. This threshold initially flagged 28,717 candidates. After strict manual verification to remove false positives, we ultimately identified 25,448 photographic-style design patents and excluded them from the ArcFace training and validation pool.

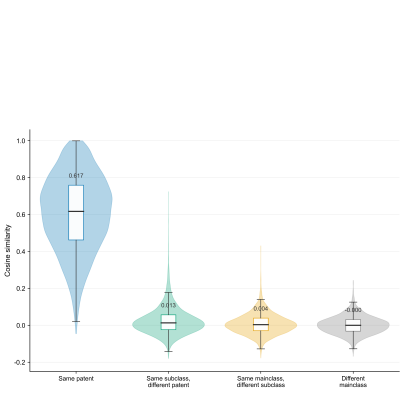


(a) Case 1 (ID: D0823510): query expansion promotes additional same-patent views.

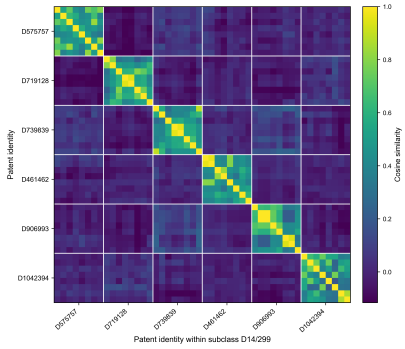


(b) Case 2 (ID: D0830028): query expansion suppresses visually similar distractors.

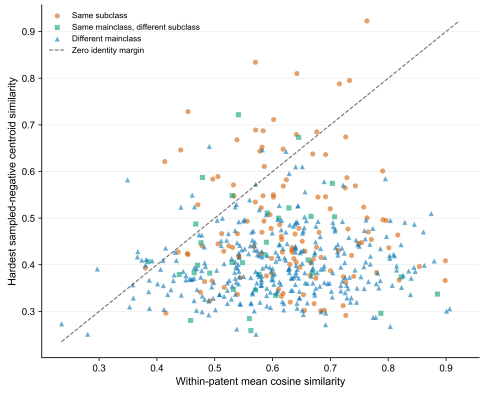
Figure 4: Two qualitative query-expansion examples. The baseline rows use raw cosine similarity, while the bottom rows use query expansion ($k = 3$, $\alpha = 1.0$) to promote same-patent views and suppress visually similar distractors. Solid boxes indicate correct matches (same patent ID), whereas dashed boxes indicate distractors (different patent IDs).



(a) Pair-level distributions.



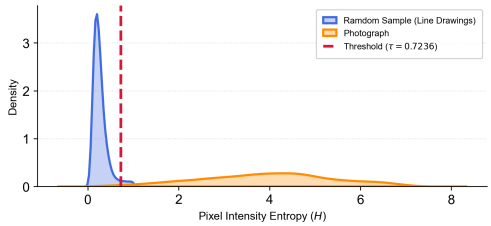
(b) Within-subclass heatmap.



(c) Hard-negative identity margins.

Figure 5: Direct cosine-space analysis of the validation embeddings. (a) Same-patent pairs have much higher similarity (median 0.617) than the three different-patent groups. (b) Within D14/299, same-patent views form bright diagonal blocks despite a shared subclass. (c) Against up to 100,000 candidate patents from the broader corpus, many plotted validation identities retain a positive margin (below the dashed line), while a smaller set remains difficult.

(a) Photographic-style images.



(b) Entropy distributions.

Figure 6: Entropy-based photograph filtering. (a) Representative photographic-style images. (b) Pixel-intensity entropy distributions for line drawings (blue) and photographs (orange). The red dashed line ($\tau = 0.724$) marks the filtering threshold.

References

[1] Kehinde Ajayi, Xin Wei, Martin Gryder, Winston Shields, Jian Wu, Shawn M. Jones, Michal Kucer, and Diane Oyen. 2023. DeepPatent2: A Large-Scale Benchmarking Corpus for Technical Drawing Understanding. *Scientific Data* 10, 1 (2023), 772.

[2] Song Bai, Xiang Bai, Qi Tian, and Longin Jan Latecki. 2017. Ensemble Diffusion for Retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, Italy, 774–783.

[3] Hui-Ching Chuang. 2025. *Final Technical Report: Design Patent Research Topics*. Technical Report NSTC 112-2410-H-305-080-MY2. Department of Statistics, National Taipei University, New Taipei City, Taiwan. Project period: 2023-08-01 to 2025-07-31; report date: 2025-10-28.

[4] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. 2007. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Rio de Janeiro, Brazil, 1–8.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Long Beach, CA, USA, 4690–4699. doi:10.1109/CVPR.2019.00482

[6] Michael Donoser and Horst Bischof. 2013. Diffusion Processes for Retrieval Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Portland, OR, USA, 1320–1327.

[7] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2016. Deep Image Retrieval: Learning Global Representations for Image Search. In *Computer Vision – ECCV 2016*. Springer, Amsterdam, The Netherlands, 241–257.

[8] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. 1973. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3, 6 (1973), 610–621.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 770–778.

[10] Kotaro Higuchi and Keiji Yanai. 2023. Patent Image Retrieval Using Transformer-Based Deep Metric Learning. *World Patent Information* 74 (2023), 102217. doi:10.1016/j.wpi.2023.102217

[11] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 2077–2086.

[12] Shuo Jiang, Jianxi Luo, Guillermo Ruiz Pava, Jie Hu, and Christopher L. Magee. 2020. A Convolutional Neural Network-Based Patent Image Retrieval Method for Design Ideation. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 83983. American Society of Mechanical Engineers, Virtual Conference, V009T09A039.

[13] Jagan N. Kapur, Prasanna K. Sahoo, and Andrew K. C. Wong. 1985. A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram. *Computer Vision, Graphics, and Image Processing* 29, 3 (1985), 273–285.

[14] Alla Kravets, Nikita Lebedev, and Maxim Legenchenko. 2017. Patents Images Retrieval and Convolutional Neural Network Training Dataset Quality Improvement. In *Proceedings of the IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine" (ITSMSSM 2017)*. Atlantis Press, Tomsk, Russia, 287–293.

[15] Michal Kucer, Diane Oyen, Juan Castorena, and Jian Wu. 2022. DeepPatent: Large Scale Patent Drawing Recognition and Retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, HI, USA, 2309–2318.

[16] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 11976–11986.

[17] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, New Orleans, LA, USA, 8 pages.

[18] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

[19] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, and Ganesh Venkatesh. 2018. Mixed Precision Training. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, Vancouver, BC, Canada, 12 pages.

[20] Claude E. Shannon. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423.

[21] Homaira Huda Shomee, Zhu Wang, Sathya N. Ravi, and Sourav Medya. 2024. IMPACT: A Large-Scale Integrated Multimodal Patent Analysis and Creation Dataset for Design Patents. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 37. Curran Associates, Inc., Vancouver, BC, Canada, 125520–125546.

[22] Homaira Huda Shomee, Zhu Wang, Sathya N. Ravi, and Sourav Medya. 2025. A Survey on Patent Analysis: From NLP to Multimodal AI. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vienna, Austria, 8545–8561. doi:10.18653/v1/2025.acl-long.419

[23] Stefanos Vrochidis and Ioannis Kompatsiaris. 2010. Towards Content-Based Patent Image Retrieval: A Framework Perspective. *World Patent Information* 32, 2 (2010), 94–106.

[24] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. 2023. ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Vancouver, BC, Canada, 16133–16142.