Materialistic: Selecting Similar Materials in Images

PRAFULL SHARMA, MIT, USA and Adobe Research, USA
JULIEN PHILIP, Adobe Research, UK
MICHAEL GHARBI, Adobe Research, US
BILL FREEMAN, MIT, US
FREDO DURAND, MIT, US
VALENTIN DESCHAINTRE, Adobe Research, UK

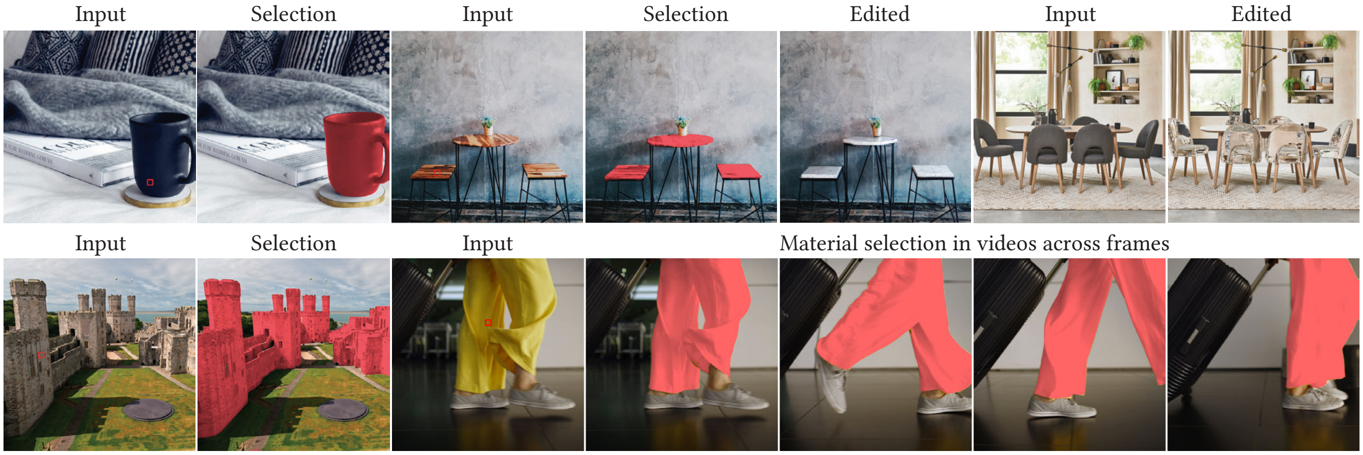

Fig. 1. Given an input image and a pixel selection (marked with a red square), our method automatically selects all the pixels that have the same material as the query. Our algorithm can identify materials shared by different objects (table and stools), and is robust to shading variations (castle example). Material-based selection enables downstream applications, such as material editing or replacement (top right). Our approach can be extended to select materials across multiple images, enabling material selection in videos (bottom right) *without* requiring any optical flow propagation.

Separating an image into meaningful underlying components is a crucial first step for both editing and understanding images. We present a method capable of selecting the regions of a photograph exhibiting the same material as an artist-chosen area. Our proposed approach is robust to shading, specular highlights, and cast shadows, enabling selection in real images. As we do not rely on semantic segmentation (different woods or metal should not be selected together), we formulate the problem as a similarity-based grouping problem based on a user-provided image location. In particular, we propose to leverage the unsupervised DINO [Caron et al. 2021] features coupled with a proposed Cross-Similarity Feature Weighting module and an MLP head to extract material similarities in an image. We train our model on a new synthetic image dataset, that we release. We show that our method generalizes well to real-world images. We carefully analyze our model's behavior on varying material properties and lighting. Additionally, we evaluate it against a hand-annotated benchmark of 50 real photographs. We further demonstrate our model on a set of applications, including material editing, in-video selection, and retrieval of object photographs with similar materials. Project website: https://prafullsharma.net/materialistic/

CCS Concepts: • **Computing methodologies → Rendering**.

Additional Key Words and Phrases: material, selection, segmentation

ACM Reference Format:
Prafull Sharma, Julien Philip, Michael Gharbi, Bill Freeman, Fredo Durand, and Valentin Deschaintre. 2023. Materialistic: Selecting Similar Materials in Images. *ACM Trans. Graph.* 42, 4, Article 1 (August 2023), 14 pages. https://doi.org/10.1145/3592390

1 INTRODUCTION

In this work, we present a method to select image regions with the same material as a given query pixel. This enables a wide variety of image editings as shown in Figure 1, and can be used to guide down stream-tasks such as inverse rendering [Nimier-David et al. 2021]. Material selection is a challenging ill-posed problem because a material's appearance can vary drastically within a single image, depending on the viewing angle and the local illumination, such that two pixels with the same material can exhibit very different reflected colors and intensities. Since a pixel's color is a complex

Authors' addresses: Prafull Sharma, MIT, USA, Adobe Research, USA, prafull@mit.edu; Julien Philip, Adobe Research, UK, juphilip@adobe.com; Michael Gharbi, Adobe Research, US, mgharbi@adobe.com; Bill Freeman, MIT, US, billf@mit.edu; Fredo Durand, MIT, US, fredo@mit.edu; Valentin Deschaintre, Adobe Research, UK, deschain@adobe.com.

0730-0301/2023/8-ART1 $15.00
https://doi.org/10.1145/3592390

function of the scene's geometry, illumination, and materials, the converse can also hold: two pixels with the same radiance, may belong to different materials. Yet, humans can identify objects that share the same material with surprising accuracy, regardless of an object's shape, and despite shading variations and strong light-dependent effects, such as specular reflections and cast shadows. It is remarkable, for instance, that we can, from a single image, identify that both chairs and the table in Fig 1 are made of the same wooden material.

In contrast to semantic segmentation, our method does not rely on a predetermined closed set of material classes. Instead, it dynamically evaluates material similarities between a user query location and all other pixels. This approach generalises to materials which have not been seen during training. We further show our method is robust to variations in shading, and in the geometries on which the material appears.

In this paper, we consider that two surfaces have the same material if they share the same texture and reflectance properties. For instance, we consider a wood with growth color variations, or a wallpaper with small repeating patterns to be single materials. However, we consider two woods with different grain textures, or different colours to be distinct materials.

To the best of our knowledge, ours is the first method that can select image regions based on the material at a user-selected pixel. Existing selection tools either perform selections based on color or intensities, requiring continued user interactions (e.g., the "lasso" tool), or perform object-level selections, using semantic and instance-level segmentation models [Cao et al. 2020; He et al. 2017; Tan and Le 2019; Wang et al. 2020b,a; Yuan et al. 2018]. Color-based methods and texture segmentation methods [Belongie et al. 1998; Chen et al. 2013; Deng and Manjunath 2001; Haindl and Mikes 2008; Todorovic and Ahuja 2009] are not robust to shading variations, as shown in Figure 8. Existing material segmentation approaches [Bell et al. 2015; Upchurch and Niu 2022] do not provide sufficient granularity: they are limited to a fixed set of high-level, predefined material classes (e.g., wood vs. metal). This precludes selecting a specific wood material in a scene containing distinct types of wood for example. Object selection methods built on instance-level segmentation are closer in spirit to our approach, but cannot perform precise material selections since a single object can be made of several materials, and a given material can appear on multiple objects.

To perform in-the-wild natural image material selection, we use a pre-trained self-supervised vision transformer, DINO [Caron et al. 2021], as a fixed feature extractor to compute a patch-level representation of the input image, leveraging its natural image priors. We then specialize these generic pretrained features for material selection using a multi-scale neural network. To specify the query pixel, we propose a Cross-Similarity Feature Weighting mechanism that modulates features at different resolutions and fuses them to obtain a material similarity score. We train our model exclusively on a synthetic dataset containing 50,000 images of indoor scenes rendered using a physically-based path tracer. The images were rendered using 100 indoor scenes with defined camera trajectories and 16,000 physically-based rendering materials.

Despite being trained on synthetic indoor scenes, we show that our model exhibits great generalization to real photographs, including outdoor images. Further, our method supports cross-image selections: a query embedding from a given image can be used to select similar materials in other images. This enables material selection in videos or material-based image retrieval in object picture databases. We demonstrate these applications and analyze the behavior of the method through a set of controlled experiments that vary material, color, lighting, selection position, and image resolution.

In summary, we propose a method that adds to the palette of image selection tools, simplifies a wide range of editing tasks, and provides important information for downstream tasks like material recognition and acquisition. We enable this through the following key contributions:

- The first material selection method for natural images, robust to shading and geometric variations.
- A novel, query-based, architecture inspired by vision transformers, allowing to select pixels based on user input.
- A new large dataset of photorealistic synthetic HDR images with per-pixel fine-grained material labels.

2 RELATED WORK

Semantic segmentation. Parsing a scene into "things and stuff" [Adelson 2001] is critical for scene understanding. It has led to the development of semantic and instance segmentation datasets with per-pixel annotations of object classes [Caesar et al. 2018; Cordts et al. 2016; Everingham et al. 2015; Geiger et al. 2012; Lin et al. 2014; Zhou et al. 2017]. Fully convolutional networks [Long et al. 2015] have become the standard for image segmentation, with ever-improving efficiency and accuracy on established benchmarks [Cao et al. 2020; Chen et al. 2017; He et al. 2017; Tan and Le 2019; Wang et al. 2020a]. Recent methods have also explored open vocabulary image segmentation [Ghiasi et al. 2021]. Although critical steps towards scene understanding, these methods are often limited by to the fixed set of labels or vocabulary they use during training and cannot adjust their segmentation for previously unseen labels. In contrast, our method can dynamically adapt its selection to a user query.

Material classification. Material classificaton is a long standing problem [Leung and Malik 2001]; it aims at recognising the type of material in an image based on a pre-defined set of classes. Prior to deep learning, methods relied on various filter banks [Fogel and Sagi 1989; Leung and Malik 2001] to extract relevant features for classification. Based on improvements in deep learning for image segmentation, per-pixel classification architecture were proposed for material types (metal, wood, ...) [Bell et al. 2015; Cimpoi et al. 2014; Schwartz and Nishino 2013, 2016] and material properties, such as fuzziness [Schwartz and Nishino 2020]. Specialized angular imaging capture systems were also proposed to further improve automatic material classification in the wild [Xue et al. 2022]. Unlike ours, these approaches are limited to a pre-defined set of material classes, and therefore cannot handle materials outside their label set. Further, they group different variations of a material in the same generic class (e.g. ‘wood’), despite strong intra-class appearance variations.

Material segmentation. Prior work has also extensively studied texture segmentation using co-occurrence matrices [Haralick et al. 1973], EM [Belongie et al. 1998], filtering [Randen and Husoy 1999; Reyes-Aldasoro and Bhalerao 2006], and watershed [Malpica et al. 2003]. These methods can segment contiguous texture regions, but do not handle disjoint regions, e.g., when multiple objects have the same material, and they do not enable a user input to specify the selection. Flat surface material segmentation typically relies on Matrix Factorization [Lawrence et al. 2006] or scribble interfaces, letting user control the segmentation [Chen et al. 2013; Hu et al. 2022b; Lepage and Lawrence 2011; Pellacini and Lawrence 2007]. These approach are however limited to the BTF/SVBRDF domain and cannot handle natural images. Different methods were proposed for scribble-based natural image decomposition into intrinsic images [Bousseau et al. 2009], or for color and local statistics based image editing [An and Pellacini 2008]. Their assumptions are too limiting for our material selection task.

Attention models and Vision Transformers. Recently, the attention mechanism [Vaswani et al. 2017] has been used in several vision tasks, such as image classification [Chen et al. 2020; Dosovitskiy et al. 2020; Hu et al. 2018], semantic segmentation [Ranftl et al. 2021; Wang et al. 2021, 2020b], super-resolution [Cao et al. 2021; Chen et al. 2022; Lu et al. 2021; Yang et al. 2020], image generation [Peebles and Xie 2022; Zhang et al. 2022], and self-supervised visual representation learning [Caron et al. 2021; Deng and Manjunath 2001]. Specifically, DINO [Caron et al. 2021] uses a Vision Transformer (ViT) to performs self-distillation to learn visual representations and demonstrate unsupervised class-specific salient object segmentations. Moreover, STEGO [Hamilton et al. 2022] uses DINO as the backbone visual representation to extract dense semantic correspondence between images and semantic segmentations. Likewise, we build our material selection model atop pre-trained DINO features, benefiting from their large-scale real image prior. Also, our Cross-Similarity Feature Weighting layer injects the user’s pixel query into our model using a cross-similarity weighting scheme inspired by cross-attention [Vaswani et al. 2017].

Material inverse rendering. Inverse rendering for materials aims at recovering appearance properties of materials from image(s) [Guarnera et al. 2016]. This problem is inherently ill-posed. So, several methods rely on data-learned prior [Deschaintre et al. 2018, 2019, 2021; Gao et al. 2019; Guo et al. 2021, 2020; Li et al. 2020, 2018a,b; Zhou et al. 2022]. Other methods rely on a stationarity assumption [Aittala et al. 2016, 2015; Deschaintre et al. 2020; Henzler et al. 2021] to extract material information from a few images. Inverse rendering methods [Azinović et al. 2019; Nimier-David et al. 2021] often explicitly require material segmentation as an input. Hu et al. [2022a] use it to improve their material editing results. Our material selection approach is orthogonal to these works. It can be used as guidance to these methods, to better share information across the image and facilitate inverse rendering.

Material datasets. Multiple datasets providing some level of material information have been proposed for classification, segmentation [Bell et al. 2013, 2015; Liu et al. 2010; Schwartz and Nishino 2019; Upchurch and Niu 2022], material inverse rendering [Deschaintre et al. 2018] or image editing tasks such as relighting [Griffiths et al. 2022; Murmann et al. 2019; Nicolet et al. 2020; Philip et al. 2019, 2021]. However, these datasets do not contain fine per-pixel material annotations. Some of them do contain per-pixel material *class* information (*i.e.* wood, metal), but they do not differentiate between intra-class instances. The lack of fine-grained material annotation creates false positives for our task (two different woods would be in the same class) and prevents us from using these datasets. As a result, we chose to render a new synthetic dataset containing 50, 000 images of indoor scenes and per-pixel material segmentation which respects the intra-class variations of materials.

Selection tools. Typical selection tools (e.g., those found in Photoshop) include the lasso or "magic wand", based on color information, as well as object-based selection tools, built on semantic and instance segmentation technologies. Color-based selection tools are insufficient for material selection, because lighting and geometric variations can significantly alter the reflectance of a single material throughout a natural scene. Semantic and instance segmentation methods have a different goal: selecting entire objects. In our material selection task, multiple objects can have the same material (e.g., multiple plastic chairs in a conference room), and a single object can be made of multiple materials (e.g., a chair with metal legs and leather seat). Our method therefore adds a new dimension to image selections, allowing a user to easily select similar materials in an image.

3 METHOD

Given an input image and a query pixel location, our algorithm computes a scalar score that quantifies how similar materials at each pixel of the input image is to the material at the query pixel, from which we derive a binary material selection mask by thresholding. The user can refine the selection by tweaking the threshold.

Our method, illustrated in Figure 2, starts from rich self-supervised natural image features from a pretrained vision model (§ 3.1), atop which we train a material-aware multi-scale encoder (§ 3.2). This encoder serves two purposes. First, it specializes the representation to be sensitive to material properties and insensitive to lighting, objectness or other discriminative properties the pretrained features may contain. Second, it lets us refine the spatially coarse pretrained features into more precise per-pixel features. We inject the spatial query point using a novel feature aggregation mechanism that weights the encoder’s internal features by cross-similarity at each scale of the encoder (§ 3.2.2). We then fuse the multi-scale information before computing the final, per-pixel material similarity score (§ 3.2.3). Because of the lack of high-quality, publicly available datasets with fine-grained material annotations, we train our encoder on a new large dataset of photorealistic renderings of interior scenes with ground-truth material labels (§ 3.3). Using features from a large pretrained vision model trained on natural images together with synthetic training data with ground-truth label gives us the best

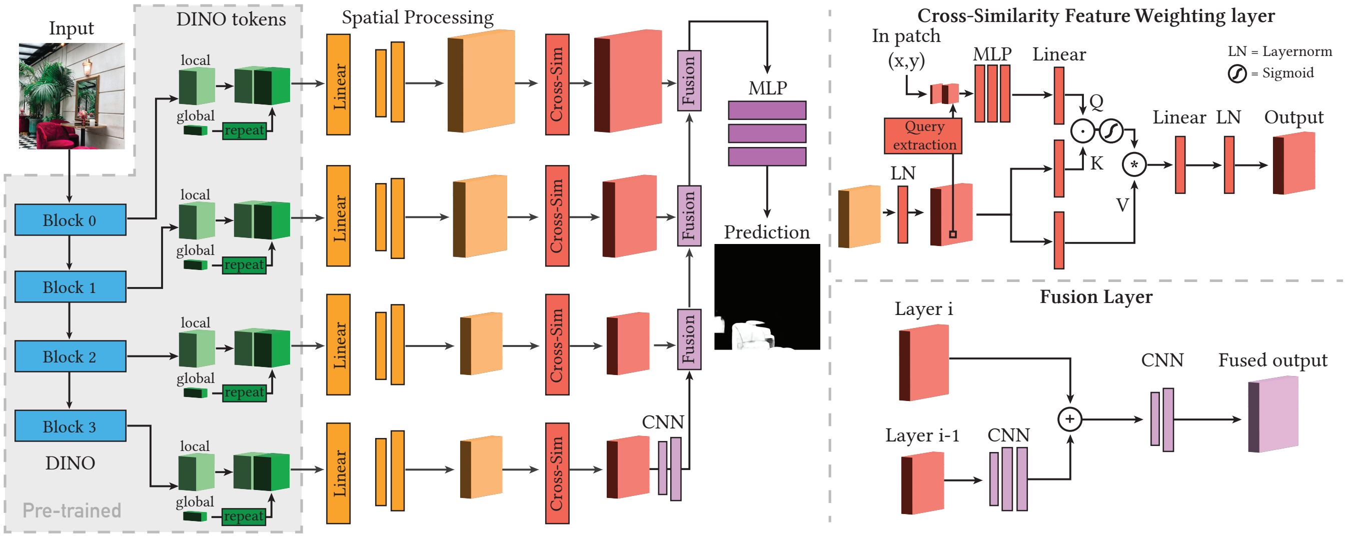

Fig. 2. **Model Architecture.** We illustrate our model architecture in this figure. The DINO features extracted on the left come from a frozen, pre-trained DINO model. We then process these layers independently at different resolution, injecting the user reference through our Cross-Similarity Feature Weighting layer (Cross-Sim) before fusing the features and extraction the similarity prediction through a fully-connected head.

of both worlds: labels that are cheap to acquire, and straightforward generalization to natural images. We provide the details of our network architectures in supplemental material.

3.1 Pre-trained DINO features

Self-supervised vision transformers (ViT) features have recently been shown to encode remarkable properties, such as rich semantic segmentation information [Caron et al. 2021], which makes them a natural starting point for our material selection task. As we discuss later (see Section 3.3), starting from features pretrained on natural images has the added advantage of mitigating the real–synthetic domain gap that often arises when training neural networks purely on synthetic data.

Concretely, we process our input image using DINO's ViT 8 × 8 configuration [Caron et al. 2021] and extract a subset of its intermediate feature tensors. Internally, DINO splits the image into non-overlapping 8 × 8 patches, called 'tokens', which are then processed by a series of transformer blocks. Each block maintains a set of local tokens, which encode local patch information; and a global token that encodes global context information [Caron et al. 2021; Hamilton et al. 2022]. The DINO ViT model contains 12 attention blocks, we use the outputs of four blocks at index (2, 5, 8, 11) as our starting feature representation, inspired by Ranftl et al. [2021]. We denote the local tokens, viewed as spatial feature tensors, by $\phi_i \in \mathbb{R}^{d \cdot \frac{h}{8} \cdot \frac{w}{8}}$, and the global tokens with $\psi_i \in \mathbb{R}^d$, where $h, w \in \mathbb{N}^2$ are the input's spatial dimensions, $d = 768$ is the feature dimension, and $i \in \{1, \ldots, 4\}$ indexes the blocks. Because of the transformer's tokenization, the local feature tensors have $\frac{1}{8}$ the spatial resolution of the input image.

In section 5, we show with an ablation that the DINO features significantly improve the selection quality, compared to a UNet-based baseline trained from scratch.

3.2 Material feature encoder

Despite their remarkable properties, the DINO features are generic and do not possess the invariants that make for a robust material representation. Therefore, we transform them into material-specific features (§ 3.2.1) by training a custom encoder on synthetic renderings with material ground-truth labels (§ 3.3). We then condition the features on our query selection (§ 3.2.2), to finally compute our material similarity score (§ 3.2.3).

3.2.1 *Multi-scale features.*

Our encoder operates at multiple scales, to increase the spatial resolution of the DINO features and analyse multiple scales in the selection process. We combine the global and local features and expand their spatial dimension following the pipeline of DPT [Ranftl et al. 2021].

Specifically, for each transformer block $i = 0, \ldots, 3$, we first aggregate the local and global features by replicating the global token spatially and concatenating it with the local feature tensor. Then, we process the aggregate using a convolutional network followed by a bilinear upsampling operator, with a different upscaling factor s_i for each feature block, which yields a new set of features

$$f_i(\phi_i, \psi_i) \in \mathbb{R}^{d' \times s_i \frac{h}{8} \times s_i \frac{w}{8}}, \qquad (1)$$

with $d' = 256$. We use the following upsampling factors: $s_0 = 4$, $s_1 = 2$, $s_2 = s_3 = 1$, such that earlier feature blocks are upsampled more.

3.2.2 *Query injection using cross-similarity feature weighting.*

After the convolutional stages described above, we obtained generic material features maps f_i at resolutions 1/2, 1/4, 1/8, 1/8 of the input image, respectively. To implement a dynamic selection mechanism that does not rely on a predefined set of material classes, and can generalize to materials unseen during training, we need to transform f_i into conditional features. These conditional features must account for the material at the query pixel $q \in [0,1]^2$, using normalized coordinates, to simplify the multi-scale notation.

To do so, we propose a novel cross-similarity feature weighting operator that modulates the feature at another pixel $p \in [0,1]^2$ using a query-dependent weight. We first obtain query Q, keys K and values V embeddings from f_i. K and V are computed by processing f_i with two different linear layers. To obtain Q, we first extract the embedding at location q from f_i and concatenate the in-patch pixel coordinate of the query selection to it. This provides our network with spatial information finer that the DINO patch index. We then feed this embedding to an MLP that outputs Q. The query-dependent weight of pixel p at each resolution i is then given by:

$$w_{i,pq} = \sigma(Q^T K/\sqrt{d}) \in [0,1]^{s_i \frac{h}{8} \times s_i \frac{w}{8}}, \tag{2}$$

where σ is a sigmoid activation. Given this weight, we compute the weighted multiscale features to be fused as

$$g_{i,pq} = w_{i,pq} \cdot V. \tag{3}$$

Our feature weighting scheme is inspired by the attention mechanism [Vaswani et al. 2017], with a couple differences. First, our similarity implements a one-to-many comparison, unlike the many-to-many relationship in traditional attention. Second, we do not seek to compute relative importance in the feature map, but rather a non-negative similarity score between the query and all other embeddings. So, the weights need not sum to one over the spatial dimensions, hence the use of a sigmoid in Eq. (2) instead of the usual softmax.

3.2.3 Multi-scale fusion and final material similarity score. We progressively fuse the information from our query-conditioned multi-scale features g_i from coarse to fine, using a residual network followed by 2× bilinear upsampling between each consecutive scale, until we reach the full image resolution $h \times w$, Finally, we compute our material similarity score in $[0,1]$ from the fused features using a pointwise neural network followed by a sigmoid activation.

3.3 Datasets with fine-grained annotations

As noted in Section 2, existing material datasets [Bell et al. 2013; Murmann et al. 2019; Upchurch and Niu 2022] with per-pixel material annotations contain *semantic* material annotations. These are too coarse for our application; they do not account for intra-class material variations. For instance, two different wood types share the same label. This prevents us from training and evaluating on these datasets for our task. Instead, we rendered a synthetic dataset for training, and manually annotated a dataset of 50 real images for evaluation.

3.3.1 Real-world evaluation benchmark. For evaluation, we manually annotated 50 images using Label Studio [Tkachenko et al. 2021], exhaustively segmenting on average 2 materials per image. The test images were sourced from Pixabay and Pexel and are available in the supplemental materials. They contain challenging cases such as multiple objects with the same material and objects made of multiple materials. We will release this benchmark publicly.

3.3.2 Synthetic training data. Our synthetic training dataset is composed of 50, 000 HDR images rendered using Blender Cycles [2018]. For the geometry of our scenes, we use a subset of 100 scenes from the Archinteriors collection [eve 2021]. For each of the 100 scenes, we use a camera trajectory spline and an associated viewing direction spline that was recorded using a first-person exploration of the scene for about one minute. We randomly sample camera locations and associated directions from the path for each scene. To render each image in the dataset, we randomly sample a camera position and field-of-view on a pre-defined, scene-specific camera path manually created using splines in Blender [2018]. For each rendering, we replace each material in the scene by randomly sampling a new one from a set of 3, 000 Adobe Substance Source materials which we curate to be stationary. We keep the original material assignment of each object constant, meaning objects that share the same materials in the original scene still share the same material after replacement.

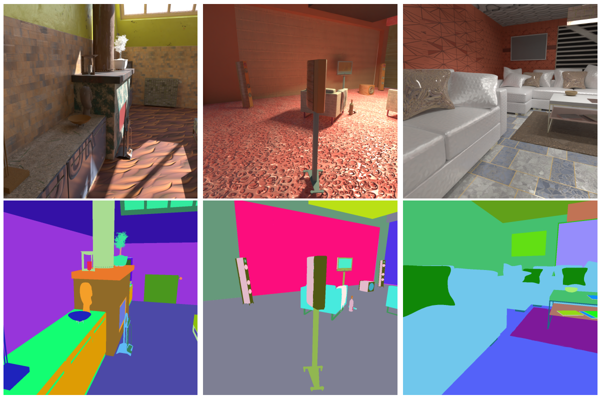

Fig. 3. **Our synthetic dataset.** We show samples from our synthetic training dataset. Top: three sample rendered images. Bottom: The corresponding material id maps, ids are mapped to random colors.

We show a few samples from our dataset in Fig.3. The random combination of material and geometries dramatically increases the diversity of our dataset. Replacing materials allows us to render an accompanying material ID map where the IDs are global across the dataset. By using such ID maps for supervision, we implicitly define *similar materials* as materials sharing the same stationary SVBDRF. While we do not use the cross-image consistency of the IDs during training, we show in Figure 5 that our method is still capable making selections across multiple images. We render at a 1024 × 1024 resolution with 256 samples per pixel using the original scene lighting. Rendering the full dataset takes about 24 hours using 8 NVIDIA A10G GPUs. Upon publication, we will release the 50, 000 HDR images and material ID pairs to facilitate further research on material selection.

3.4 Implementation details

We train our model for 30 epochs on the dataset described in Section 3.3, using the Adam optimizer with a learning rate of 10^{-4} on 2 V100 GPUs, with a batch size of 8 images per GPU, following the distributed data-parallel (DDP) training approach. During training, we apply random exposure, saturation, and brightness augmentations to a random 512 × 512 crop of our training renderings. At inference time, our model computes material similarity in a 512 × 512 image in 240 ms on a V100 GPU.

Loss function. Given a query pixel, we compute a binary cross-entropy loss to measure whether the predicted selection (with threshold 0.5) matches the desired material regions in the image, specified by the query pixel location and the ground truth material segmentation of our synthetic data.

Selection refinement using KNN matting. Because of the low resolution of the DINO feature, our approach may not always select the boundaries of thin features perfectly. We found that applying the KNN-Matting algorithm [Chen et al. 2013] on the edges of our selection was sufficient to improve the resolution of small features, if desired. When using this post-processing, we automatically define the positive and negative anchors required by KNN matting, by eroding and dilating our selection mask with a $(9, 9)$ kernel, marking the content of the erosion as positives, and the inverse of the dilation as negative anchors. *Unless explicitly specified, we show the direct output of our network and do not apply the refinement step discussed in here, which is separately illustrated in Figure 8 and in the supplemental material.*

4 RESULTS

In this section, we present the qualitative results from our model, and its ability to perform material selection across images allowing us to directly apply our method to selection in high-resolution images and video frames for example.

4.1 Qualitative results

In Figure 4, we present results obtained by directly applying our method to a subset of our evaluation dataset. This shows our model is robust to strong shading variations (first and second columns), to the presence of different objects sharing the same material (fifth and seventh columns), and to surface orientation (seventh column). We also show in Figure 8 (last column) that the KNN refinement step discussed in Section 3.4 can slightly improve selection boundaries and challenging thin structures.

4.2 Cross-image selection

A natural extension of our method is the selection of similar materials across images. Given an image in which a user provides the query, we show that we can select similar materials in different images, as long as the lighting does not vary dramatically (we evaluate the robustness to lighting variation in Figure 10).

Given an input image I_{query}, used to define the user-query, we want to select similar materials in a different image I_{select}. To do so, we process both I_{query} and I_{select} independently up until the Cross-Similarity Feature Weighting layers. We then compute the query (Q in Figure 2) using the features from I_{query} while we use the spatially processed images features (K and V in Figure 2) from I_{select}. The cross-similarity features are then fused in the same way than for selection in a single image.

We show cross-image selection results in Figure 5. We can see that despite varying viewpoints and lighting conditions, our method can select similar materials across images. The "Arc de Triomphe" in the first column also illustrates that our method generalises well to outdoor images, further demonstrated in the video results discussed in Section 4.3, despite the training data being confined to indoor renderings. Note that even though the two images have different scene lighting, the model outputs robust selections. On the second and third column we observe that the materials from the chairs and sofa are well identified in the target image even though orientation and lighting vary.

Table 1. **Quantitative metrics.** Mean IoU scores of all models evaluated on our densely annotated material dataset containing 50 real images. We compute the mIoU for 10 randomly selected user-query pixel for each image and average the results. –see supplemental. We also provide a tally which indicated if the method requires negative samples during inference.

Model	Negative Samples	mIoU ↑
KNN matting (3 patches)	✓	0.617
KNN matting (5 patches)	✓	0.677
KNN matting (3 patch, albedo estimates)	✓	0.567
KNN matting (5 patch, albedo estimates)	✓	0.640
DMS [Upchurch and Niu 2022]		0.38
UNet on RGB		0.612
DINO ViT16 backbone		0.877
(Ablation) Single Dino Block		0.5
(Ablation) No Cross-Sim layer		0.9
(Ours) DINO ViT8 backbone		0.917
Ours refined with KNNmatting		**0.92**

4.3 Videos

The ability to select materials across images can also be used directly on videos where the reference pixel selection is made in the first frame, and the model selects material on a per-frame basis to extract the desired material across the video. We show results in Figure 1 and 7 and supplemental materials. These videos and Figure 5 further illustrate the stability and robustness of the approach even though no temporal smoothing is applied to the output.

4.4 High resolution images

The high computational cost of self-attention to compute DINO features restricts the input to our method to 512×512 pixel resolution. However, since our model generalizes across images, this lets us evaluate it on high-resolution images by processing overlapping crops.

Specifically, we follow the same process as cross-image selection. We first obtain a query Q by running the model on a 512×512 downsampled version of the image. Then, using this query Q, we evaluate the similarity scores for a set of crops of size 512×512 using a simple sliding window with a stride of 256 pixels on the high-resolution image. For each crop, we use their respective keys K and values V.

For each pixel in the high-resolution input we average the similarity scores obtained from all crops that overlap with this pixel, which gives us our final similarity score. We present results on 1K resolution images in Figure 6.

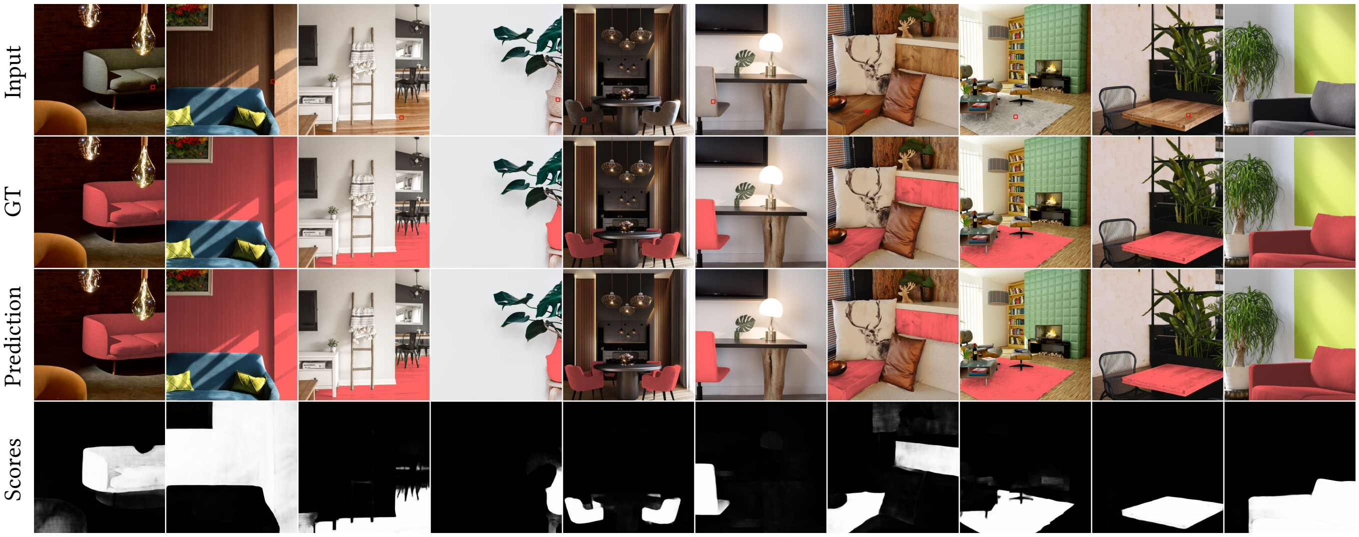

Fig. 4. **Qualitative results.** We present the input image along with the query point annotated with a red square, ground truth mask (GT), predicted mask, and the per-pixel score (top to bottom rows). Note that the user selects the center of the marked square. The prediction is the best possible mask generated by optimally thresholding the per-pixel scores. The predicted results demonstrate the robustness to shading variations (first and second columns), the presence of different objects sharing the same material (fifth and seventh columns), and surface orientation (seventh column).

5 EVALUATION

We present both quantitative and qualitative evaluations of our model through a set of ablations, comparisons and experiments to better understand its behavior and limitations. Quantitative results are shown in Table 1, in which we provide the mean Intersection over Union (IoU) score comparing the ground truth material masks and the predicted masks on our benchmark evaluation dataset for 10 randomly selected query points per image. We further show qualitative comparisons in Figure 8. As different methods require a different threshold, we always report the number with their optimal threshold selected. We perform a grid search to find the threshold that yields the highest mIOU for each method. This illustrates the best selection result that can be achieved using each method.

5.1 Ablations

Dense Material Segmentation. We compare our method to a material segmentation method [Upchurch and Niu 2022], where we run the pre-trained segmentation network on the test images and generating the binary mask as the segment that belongs to the segment at the user selection. Since this network is trained with a closed set of high-level material labels (i.e. wood, metal), the network suffers from under-segmentation as it treats all intra-class variations of the material as the same.

UNet. We explore and evaluate an alternative neural architecture. Given the image to image nature of our task, we use the fixup UNet [Ronneberger et al. 2015; Zhang et al. 2019] which has been shown to do well on single-image relighting tasks [Griffiths et al. 2022]. We train this model on the synthetic dataset presented in Section 3.3. Given an RGB input image I, the network outputs a per-pixel 32-dimensional embedding f_p. We train this approach with a binary cross entropy loss using the dot product between the query embedding f_q and all other embeddings f_p.

This approach can be seen as an extremely simplified version of our model, where the query injection happens at a single resolution, and where the dot product between query and key (which are not processed through separate MLPs) is directly used as similarity instead of being further processed. As shown in Table 1 this U-Net model achieves significantly lower mIoU, and we can see in Figure 8 that the resulting selection tends to be noisy.

Single DINO features block. Our model uses multiple blocks of the pre-trained DINO model, which we need to fuse after injecting the query, as described in Sections 3.2.2 and 3.2.3. We evaluate the performance of a simplified model that uses a single DINO feature block. For this baseline, we directly use $w_{3,pq}$ from Eq.2 as the similarity score, since no fusion is needed. Table 1 shows a single feature block with limited processing is insufficient. This confirms that the pre-trained DINO features are not natively sufficient to discriminate materials, and that our model design, which refines the 4 blocks of DINO features with a query-dependent feature combination and selection mechanism, is essential.

DINO patch size. To evaluate the impact of the patch sizes of the DINO features, we trained a variant of our model using the ViT-16 model as a backbone, using 16 × 16 patches, instead of our main method, which uses ViT-8, i.e., 8 × 8 patches. The ViT-16 backbone performs better than the more naive baselines described above. However, its lower spatial resolution leads to less precise segmentations, compared to the ViT-8. This is especially visible around material edges in Figure 8. This also leads to a lower accuracy overall (see Table 1).

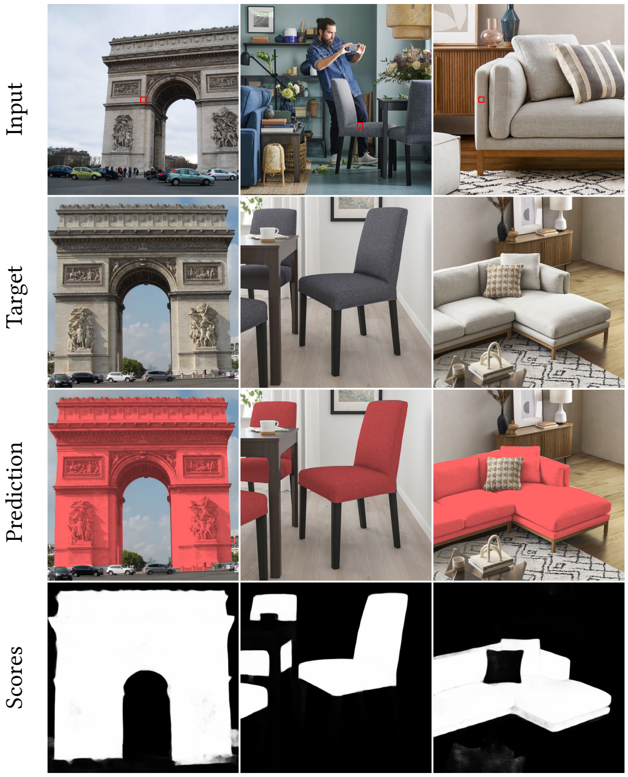

Fig. 5. **Material selection across images.** From a selection in one image (top row), our model can find the corresponding material in a second image (second row). The reference pixels are highlighted with a red square and the predicted selection and associated similarity scores are shown in the third and fourth rows. We show that our model can indeed select across images, despite the differences in environment, viewpoint, and lighting.

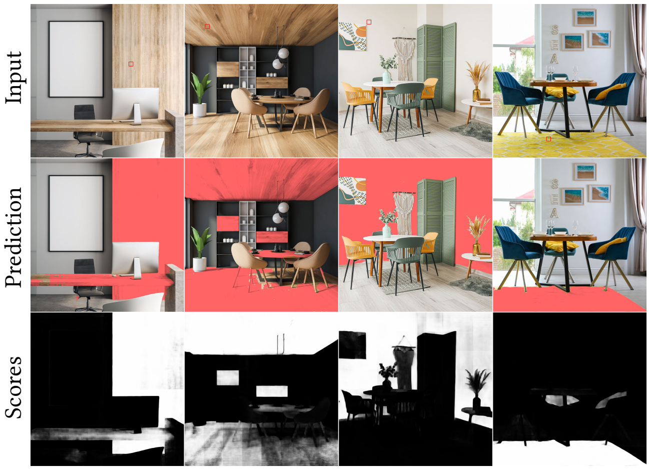

Fig. 6. **High resolution material selection.** Our method can be evaluated on higher resolution such as 1024×1024 even though the method was trained on 512×512. Using the query embedding from a downsampled version of the image, we compute the similarity to the query of crops of the high-resolution image using a sliding window with stride 256. These scores coming from different patches are averaged for each pixel in the high-resolution image.

Selection injection. Finally, we ablate our Cross-Similarity Feature Weighting layer, replacing it with a simple concatenation, at every pixel, of the query features vector with the pixel's feature vector. This new tensor is then processed by a linear layer before the fusion step. This simplified network, denoted "(Ablation) No Cross-Sim layer" in Table 1, performs reasonably well. But removing our feature reweighting and the injection of the local sub-patch position of the query pixel (Figure 2 top-right), leads to degraded spatial localization. This is especially visible in Figure 8, where the selection is imprecise around the chairs edges and some part of the TV (top-row). The baseline also overselects the wooden chair background (bottom row). More comparisons are provided in the supplemental material.

5.2 Comparisons

In addition to these ablations, we compare to existing selection methods. Our method being the first to enable material selection for natural images, we compare our model to the following selection tools: KNN matting [Chen et al. 2013] in multiple color spaces (HSV, albedo from intrinsic images[Li et al. 2020]) and the Magic Wand and Object selection tools in Photoshop. For all comparisons and ablations, we show additional results in supplemental material

KNN matting. We use the open-source implementation of KNN matting [Germer et al. 2020] on our test data in HSV space [Chen et al. 2013]. The method requires positive and negative anchors, so we report their results using 3 and 5 32 × 32 patches as positive and negative samples. We select the patch randomly within the ground-truth positive/negative regions. We also evaluate KNN matting on an albedo map extracted using intrinsic image decomposition [Li et al. 2020], to try and minimize the effect of shading in the selection.

Based on the mean IoU scores in Table 1, our method outperforms the baseline models. Since KNN matting only takes into account the observed color and the local pixel position, without any notion of lighting and geometry, its selection degrades when cast shadows and light dependent effects are present. While the albedo component extracted from an intrinsic image decomposition method should remove shading , current methods do not handle global illumination effects perfectly [Garces et al. 2022], leading to artefacts in the image. Moreover, recent intrinsic image decomposition methods output low resolution albedo maps. Together, this limits the quality of KNN matting on intrinsic images baseline, so that it is no better than running the algorithm in its original HSV space.

Photoshop selections. We also compare our method qualitatively to results obtained using existing selection tools, namely the Magic Wand and Object selection tools in Photoshop. Object selection works well but it solves a different task than ours, typically selecting a single entire chair, and not just the seat, in both examples of Figure. 8.

On the other hand, Magic Wand selection is based on color similarity and often selects incorrect areas due to shading or specular highlights. As we show in Figure 8 it does not handle well shading variations and varying lighting. Further, similar RGB color (such as the TV and lights in the first row) are also incorrectly selected.

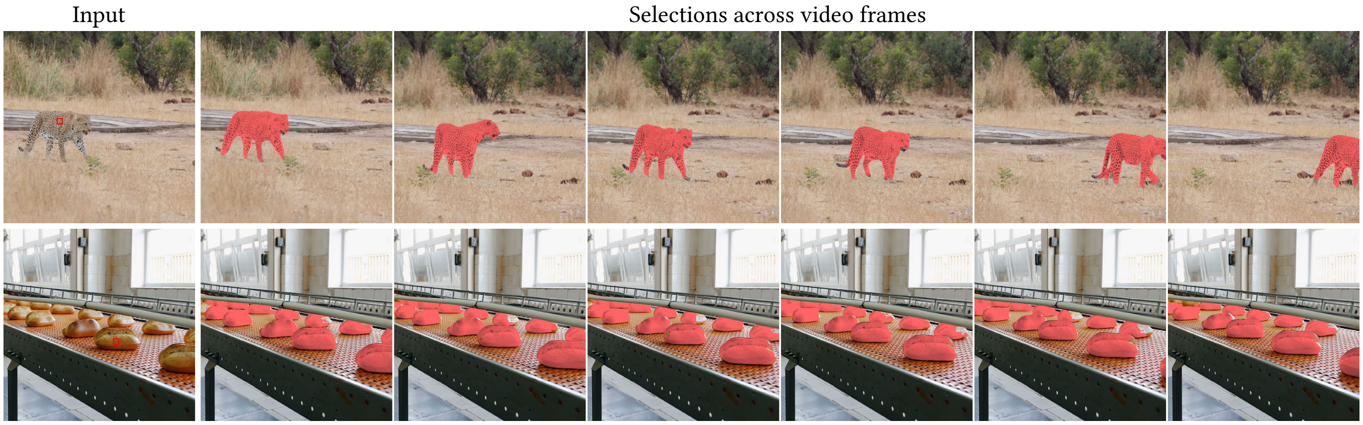

Fig. 7. **Material selection in videos.** Given a user input on the first frame of the video, our method can select the material across all frames of the video. Note that the spotted fur of the cheetah is selected in all frames even though the dry grass texture also exhibits the same low frequency statistics. The method also successfully selects all the bread rolls on the assembly row.

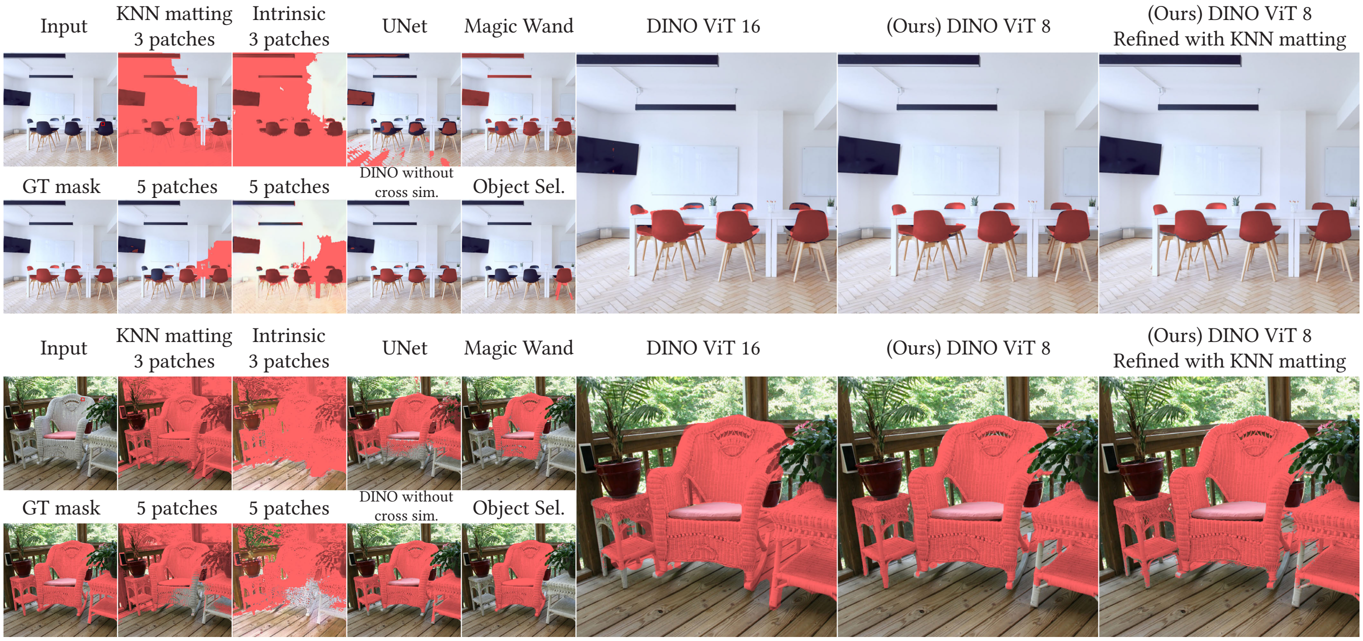

Fig. 8. **Qualitative comparisons.** We show comparisons between selected regions (marked in red) obtained with different methods. From left to right, we first show results obtained with KNN matting using a different number of positive and negative patches, and in two different color domains (HSV and Intrinsic image albedo). We then show results for our two best-performing ablations, *i.e.* the alternative UNet architecture and our architecture without the Cross-Similarity Feature Weighting mechanism. Then we present results from existing Photoshop selection tools and a VIT-16 pre-trained network. Finally, on the far right are our results and a KNN matting refined version of our results for which we refine selection borders. We can see that our method better selects the same materials in the image across different objects, despite similar colors in the image (e.g color and lamps in the top row) or the space being cluttered (bottom row). Further, the KNN Matting improved selection better follows material edges. While close, the no-cross sim ablation has more imprecise selection and overselects the TV (top-row) and the background of the wooden chair (bottom-row).

5.3 Selection consistency

We study the consistency of our model's output with respect to change in the scene lighting and selected pixel location.

Robustness to the query pixel. We evaluate the self-consistency of our model's selections by making multiple queries within a region labeled as a single material.

Figure 9 illustrates this experiment for 5 different selections marked in the input with squares of different colors along with the

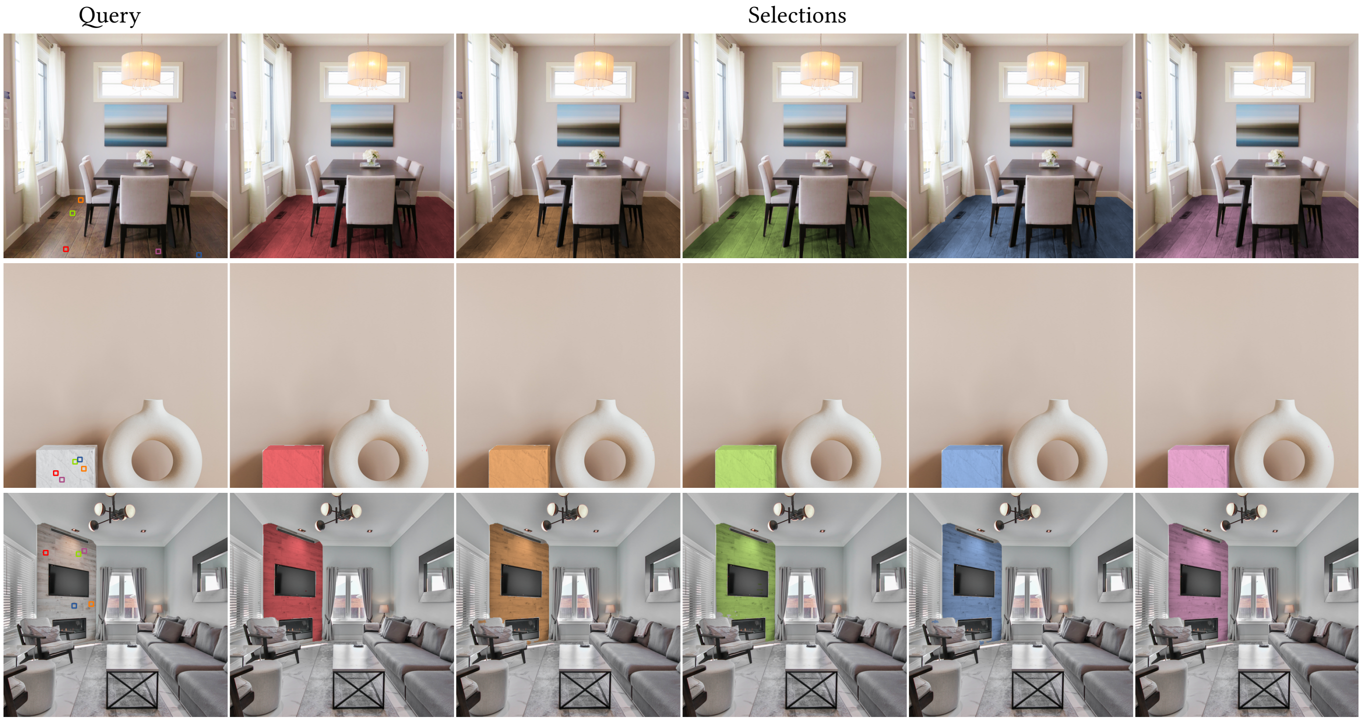

Fig. 9. **Consistency of model output.** Querying our model with 5 different query points –highlighted by a colored square– within region corresponding to the same material, the selected region is consistent. Each of the query points and the corresponding selection is presented in different colors. Note that despite the different lighting effects such as specularity and shadows around different selections, the model consistently selects the same region (row 1 and 3). Similar consistency is exhibited for selections around different regions of a spatially varying texture of marble in row 2.

corresponding output selections. Regardless location of the query pixel within a given material's region, our method's output selection is stable. In particular, as shown in rows 1 and 3, our selection does not change, even if the query points are under different illuminations. We compute cross-mIoU between the selected regions predicted with different input query points within the regions corresponding to the same material. We consider the first selection point to be the control and compute the average mIoU of 5 other selections with respect to control. We compute this over our entire benchmark dataset with each image evaluated twice with random selection of query points and obtain an average cross-mIoU of 0.9387.

Consistency under varying lighting. Previous sections showed qualitatively that our method is robust to varying lighting. Here, we evaluate this systematically. In Figure 10, we show a room photographed multiple times, from the same viewpoint, each time changing the color and intensity of the room's artificial lighting. For this test, we used the "across images" approach described in Section 4.2. The query is computed from the leftmost image in each row and then used to select the same material in the other images with the same viewpoint. Our selections are fairly robust to strong lighting variations, although slight variations can be observed in areas where the radiance changes significantly, such as the grey carpet turning deep blue in the first example.

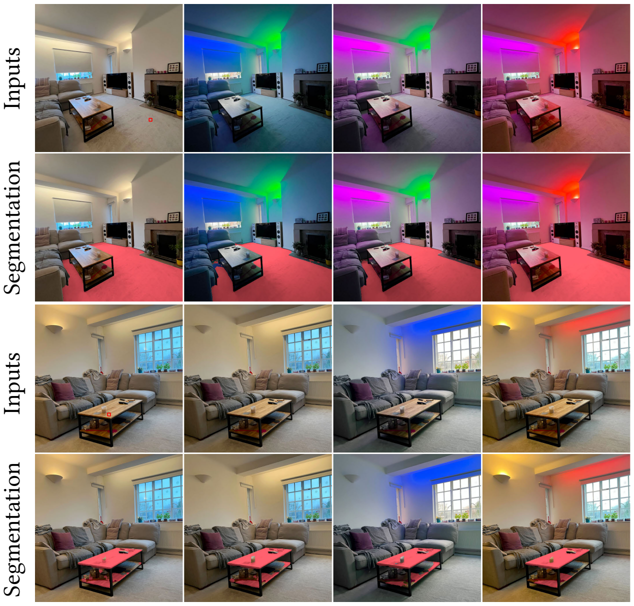

Fig. 10. **Lighting Robustness Analysis.** We show the consistency of the selection of a given material for scenes with varying lighting.

We show further examples with varying lighting from the dataset of Murmann et al. [2019] in supplemental material. We compute the cross-mIoU again (as described in §5.3) between the selections across different lighting conditions for the photographs shown in Figure 10 and examples from Murmann et al. [2019] dataset shown in the supplement material. Here we obtain an average cross-mIoU of 0.956.

5.4 Varying color and texture

Color is a very strong visual cue, on which many segmentation methods rely (e.g., Photoshop's Magic Wand or KNN Matting [Chen et al. 2013])). To evaluate the importance of color and texture in our model's selection, we designed two synthetic test cases, in which color (resp. texture) varies progressively across multiple spheres (see supplemental material.). In particular, this test helps understand what the method considers to be a single material. Our model behaves as expected, with no color variations tolerated in the selection at high threshold. As we lower the threshold, the selection is relaxed and some color variation is tolerate. Similarly, in the texture interpolation test, we see that closer interpolated textures are selected first, with a loose threshold.

5.5 Refining selections with multiple query points

To further empower artists, in our interactive demo, we allow users to select multiple positive and negative query points. The resulting score map for positive query points are combined by taking the maximum of the individually predicted similarity scores for each pixel, and thresholded with a user defined value in $[0, 1]$. Similarly, the user can select negative points to remove regions from the current selection. The predicted scores corresponding to all negative samples are also combined by computing a per-pixel maximum across all predicted scores, and then thresholded by the user using a separate threshold value. The intersection of the negative mask with the mask computed using positive query points is removed from the final selection. We illustrate this workflow in a video in supplemental material.

6 APPLICATIONS

We showcase two applications of our model: image editing and material-driven web recommendations.

Image editing. The ability to make selections based on materials opens up many image editing possibilities. For instance, Figure 11 shows examples in which we alter or replace the selected materials. In the top two rows, we modify the hue of the selection (first result column), and multiply the luminance with texture to replace the selection's appearance (second result). For the next two rows, we partially re-implement the material-editing method from Khan et al. [2006] using contemporary techniques, such as monocular depth estimation [Ranftl et al. 2022] and GAN-based inpainting for the environment [Karimi Dastjerdi et al. 2022]. The first edit for the statue (3rd row) uses their glass approximation, while the second uses a pure mirror material. In the Arc de Triomphe example (4th row), we use the glass approximation with varying roughness, implemented by blurring the environment map. The last two rows are obtained using our selection as inpainting mask in Stable Diffusion [Rombach et al. 2021] and various text prompts.

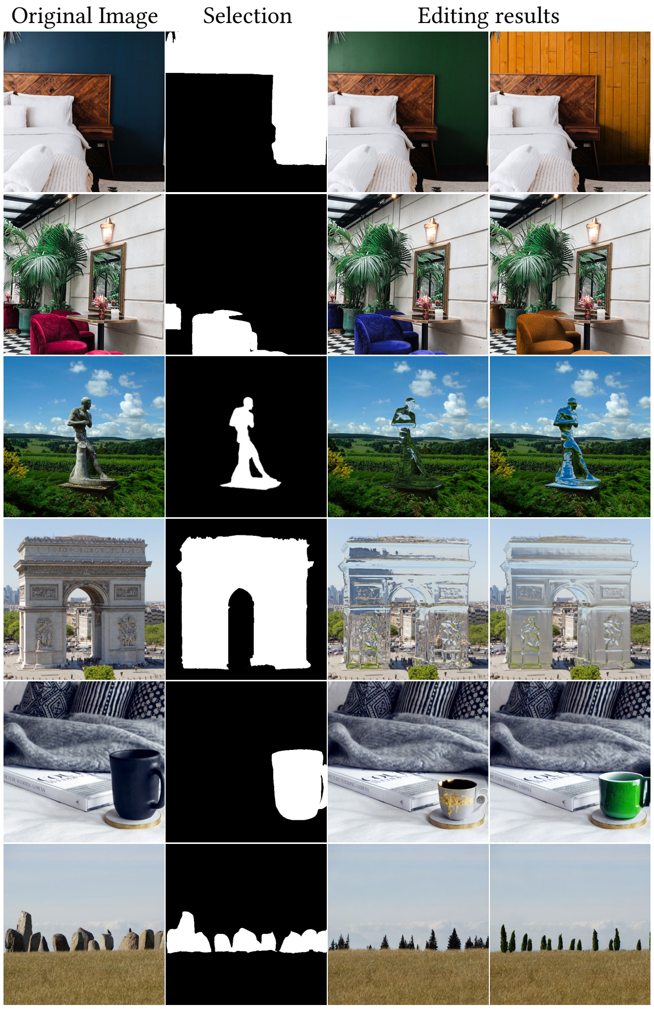

Fig. 11. **Image editing results.** Our selection method output used as input for material selection based image editing using e.g. Photoshop (top rows), Khan et al. [2006] (middle rows) and Stable Diffusion Inpainting [Rombach et al. 2021] (bottom rows). Fifth row prompts: "A white tea cup with gilding", "A green mug". Last row prompts: "Pine trees", "Mediterranean trees".

Recommendation based on a selected material. Large online datasets such as an online product catalog can be challenging to navigate. Using our method we show that we can add a new axis along which it is possible to explore the dataset: material similarity. Given a subset of 150 images for different semantic material classes (wood, plastic, leather), from the Amazon Berkeley Objects (ABO) Dataset [Collins et al. 2022], we first select an image with a material we want to see more of, and search for objects with similar material using our query embedding following the material selection across images approach (Section 4.2). We rank the images in the database's subset based on the mIoU of the selection with respect to object mask in the rest of the dataset. As shown in Figure 12, our method can

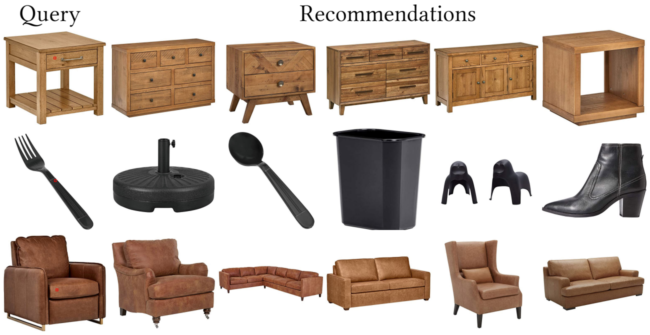

Fig. 12. **Web recommendation.** Our method can be employed to perform material based search for web recommendation. Given a query image and a user selection, we rank the images based on mIoU of the selection with respect to object mask in a random 150 images subset of the Amazon Berkeley Dataset [Collins et al. 2022] from different (wood, plastic leather). We show the complete subset in the supplemental material.

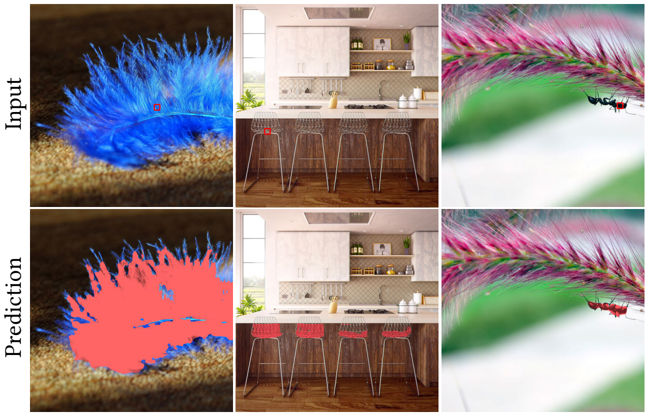

Fig. 13. **Limitations.** The first two images illustrate the challenge of selecting thin structures for our method, while the last image illustrates the difficulty of extracting precisely high frequency and small selection borders.

retrieve objects with similar materials, we show the complete subset in supplemental material.

6.1 Limitations

As shown in the evaluations, our method is robust to light and view variations. Despite being trained on purely synthetic data, it generalizes to real photographs and unseen materials, which in turns enables diverse applications.

However, as mentioned in Section 3.4 selections of fine details remain challenging. As shown in Figure 13, thin elements such as the blue feather, the thin grid on the chair or the ant are very hard to segment accurately. Our KNN refinement step helps clean the selection boundaries in difficult cases, but is insufficient to recover very thin structures. We believe stems from two limitations: the low resolution DINO features, mitigated but not entirely solved by our rescaling and feature-weighting mechanisms; and our synthetic training data, which does not contain many thin geometric structures.

At a higher level, our definition of what constitutes a single material is closely tied to the notion of material in Computer Graphics, and what artists commonly define as stationary materials. For example, we consider a wood plank as a single material, despite the wood-growth hue variations. This definition may not always align with a user's expectation, but a different definition may require more fine-grained ground truth labels. Our method is inaccurate for regions with extreme direct cast shadows, as seen in the second example presented in Fig 6. The direct cast shadows in such cases result in extremely underexposed regions revealing very little about the material in that region.

7 CONCLUSION

In summary, we propose a method for material selection in natural images. Our method builds on pre-trained generic vision features, which we specialize for material selection by training a downstream model on a new synthetic dataset. Crucially, our downstream model employs a new mechanism to merge multi-scale features and inject a user input. We demonstrate the quality and robustness of our selections on both indoor and outdoor scenes, and show it can be applied to make selections with single images, across multiple images or even in videos. We believe our method enables better high-level scene understanding and provides important information for inverse rendering optimization.

ACKNOWLEDGMENTS

This work is in part supported by NSF 1955864, Occlusion and Directional Resolution in Computational Imaging, NSF under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/) and The Toyota Research Institute. This article solely reflects the opinions and conclusions of its authors and not other entity.

REFERENCES

2021. Evermotion Arch Interior. https://evermotion.org/shop/cat/397/archinteriors.

Edward H Adelson. 2001. On seeing stuff: the perception of materials by humans and machines. In *Human vision and electronic imaging VI*, Vol. 4299. SPIE, 1–12.

Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2016. Reflectance Modeling by Neural Texture Synthesis. *ACM Trans. Graph.* 35, 4 (2016), 65:1–65:13.

Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2015. Two-shot SVBRDF Capture for Stationary Materials. *ACM Trans. Graph.* 34, 4 (2015), 110:1–110:13.

Xiaobo An and Fabio Pellacini. 2008. AppProp: All-Pairs Appearance-Space Edit Propagation. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 40, 9 pages. https://doi.org/10.1145/1399504.1360639

Dejan Azinović, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. 2019. Inverse Path Tracing for Joint Material and Lighting Estimation. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2013. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on graphics (TOG)* 32, 4 (2013), 1–17.

Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2015. Material Recognition in the Wild with the Materials in Context Database. *Computer Vision and Pattern Recognition (CVPR)* (2015).

Serge Belongie, Chad Carson, Hayit Greenspan, and Jitendra Malik. 1998. Color-and texture-based image segmentation using EM and its application to content-based image retrieval. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 675–682.

Adrien Bousseau, Sylvain Paris, and Frédo Durand. 2009. User Assisted Intrinsic Images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)* 28, 5 (2009).

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1209–1218.

Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. 2020. D2det: Towards high quality object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11485–11494.

Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. 2021. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847* (2021).

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9650–9660.

Bingling Chen, Yan Huang, Qiaoqiao Xia, and Qinglin Zhang. 2020. Nonlocal spatial attention module for image classification. *International Journal of Advanced Robotic Systems* 17, 5 (2020), 1729881420938927.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.

Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. 2013. KNN matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, 9 (Sept 2013), 2175–2188. https://doi.org/10.1109/TPAMI.2013.18

Xiangyu Chen, Xintao Wang, Jiantao Zhou, and Chao Dong. 2022. Activating More Pixels in Image Super-Resolution Transformer. *arXiv preprint arXiv:2205.04437* (2022).

Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. 2014. Deep convolutional filter banks for texture recognition and segmentation. *arXiv preprint arXiv:1411.6836* (2014).

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. 2022. ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. *CVPR* (2022).

Blender Online Community. 2018. Blender - a 3D modelling and rendering package. http://www.blender.org

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.

Yining Deng and Bangalore S Manjunath. 2001. Unsupervised segmentation of color-texture regions in images and video. *IEEE transactions on pattern analysis and machine intelligence* 23, 8 (2001), 800–810.

Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. 2018. Single-image SVBRDF Capture with a Rendering-aware Deep Network. *ACM Trans. Graph.* 37, 4 (2018), 128:1–128:15.

Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. 2019. Flexible SVBRDF Capture with a Multi-Image Deep Network. *Computer Graphics Forum* 38, 4 (2019).

Valentin Deschaintre, George Drettakis, and Adrien Bousseau. 2020. Guided Fine-Tuning for Large-Scale Material Transfer. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 39, 4 (2020). http://www-sop.inria.fr/reves/Basilic/2020/DDB20

Valentin Deschaintre, Yiming Lin, and Abhijeet Ghosh. 2021. Deep polarization imaging for 3D shape and SVBRDF acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* 111, 1 (Jan. 2015), 98–136.

Itzhak Fogel and Dov Sagi. 1989. Gabor filters as texture discriminator. *Biological cybernetics* 61, 2 (1989), 103–113.

Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. 2019. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Trans. Graph.* 38, 4 (2019).

Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, and Jorge Lopez-Moreno. 2022. A Survey on Intrinsic Images: Delving Deep into Lambert and Beyond. *International Journal of Computer Vision* 130, 3 (2022), 836–868.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3354–3361.

Thomas Germer, Tobias Uelwer, Stefan Conrad, and Stefan Harmeling. 2020. PyMatting: A Python Library for Alpha Matting. *Journal of Open Source Software* 5, 54 (2020), 2481. https://doi.org/10.21105/joss.02481

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. 2021. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143* (2021).

David Griffiths, Tobias Ritschel, and Julien Philip. 2022. OutCast: Single Image Relighting with Cast Shadows. *Computer Graphics Forum* 43 (2022).

Dar'ya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. 2016. BRDF Representation and Acquisition. *Computer Graphics Forum* (2016).

Jie Guo, Shuichang Lai, Chengzhi Tao, Yuelong Cai, Lei Wang, Yanwen Guo, and Ling-Qi Yan. 2021. Highlight-Aware Two-Stream Network for Single-Image SVBRDF Acquisition. *ACM Trans. Graph.* 40, 4, Article 123 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459854

Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. 2020. MaterialGAN: Reflectance Capture using a Generative SVBRDF Model. *ACM Trans. Graph.* 39, 6 (2020), 254:1–254:13.

Michal Haindl and Stanislav Mikes. 2008. Texture segmentation benchmark. In *2008 19th International Conference on Pattern Recognition*. IEEE, 1–4.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. 2022. Unsupervised Semantic Segmentation by Distilling Feature Correspondences. *arXiv preprint arXiv:2203.08414* (2022).

Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. 1973. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics* 6 (1973), 610–621.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

Philipp Henzler, Valentin Deschaintre, Niloy J. Mitra, and Tobias Ritschel. 2021. Generative Modelling of BRDF Textures from Flash Images. *ACM Trans. Graph.* 40, 6, Article 284 (dec 2021), 13 pages.

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. 2022a. Controlling Material Appearance by Examples. *Computer Graphics Forum* (2022). https://doi.org/10.1111/cgf.14591

Yiwei Hu, Chengan He, Valentin Deschaintre, Julie Dorsey, and Holly Rushmeier. 2022b. An Inverse Procedural Modeling Pipeline for SVBRDF Maps. *ACM Trans. Graph.* 41, 2, Article 18 (jan 2022), 17 pages. https://doi.org/10.1145/3502431

Mohammad Reza Karimi Dastjerdi, Yannick Hold-Geoffroy, Jonathan Eisenmann, Siavash Khodadadeh, and Jean-François Lalonde. 2022. Guided Co-Modulated GAN for 360 degree Field of View Extrapolation. *International Conference on 3D Vision (3DV)* (2022).

Erum Arif Khan, Erik Reinhard, Roland W. Fleming, and Heinrich H. Bülthoff. 2006. Image-Based Material Editing. *ACM Trans. Graph.* 25, 3 (jul 2006), 654–663. https://doi.org/10.1145/1141911.1141937

Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. 2006. Inverse Shade Trees for Non-Parametric Material Representation and Editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 25, 3 (July 2006).

Daniel Lepage and Jason Lawrence. 2011. Material Matting. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 1–10. https://doi.org/10.1145/2070781.2024178

Thomas Leung and Jitendra Malik. 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International journal of computer vision* 43, 1 (2001), 29–44.

Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2020. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2475–2484.

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018a. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 11207)*. 74–90.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018b. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 269.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

Ce Liu, Lavanya Sharan, Edward H Adelson, and Ruth Rosenholtz. 2010. Exploring features in a bayesian framework for material recognition. In *2010 ieee computer society conference on computer vision and pattern recognition*. IEEE, 239–246.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

Zhisheng Lu, Hong Liu, Juncheng Li, and Linlin Zhang. 2021. Efficient transformer for single image super-resolution. *arXiv preprint arXiv:2108.11084* (2021).

Norberto Malpica, Juan E Ortuño, and Andres Santos. 2003. A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters* 24, 9-10 (2003), 1545–1554.

Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. 2019. A dataset of multi-illumination images in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4080–4089.

Baptiste Nicolet, Julien Philip, and George Drettakis. 2020. Repurposing a Relighting Network for Realistic Compositions of Captured Scenes. In *Symposium on Interactive 3D Graphics and Games* (San Francisco, CA, USA) *(I3D '20)*. Association for Computing Machinery, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3384382.3384523

Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. 2021. Material and Lighting Reconstruction for Complex Indoor Scenes with Texture-space Differentiable Rendering. In *Eurographics Symposium on Rendering - DL-only Track*, Adrien Bousseau and Morgan McGuire (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20211292

William Peebles and Saining Xie. 2022. Scalable Diffusion Models with Transformers. *arXiv preprint arXiv:2212.09748* (2022).

Fabio Pellacini and Jason Lawrence. 2007. AppWand: Editing Measured Materials Using Appearance-Driven Optimization. *ACM Trans. Graph.* 26, 3 (jul 2007), 54–es. https://doi.org/10.1145/1276377.1276444

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A. Efros, and George Drettakis. 2019. Multi-View Relighting Using a Geometry-Aware Network. *ACM Trans. Graph.* 38, 4, Article 78 (jul 2019), 14 pages. https://doi.org/10.1145/3306346.3323013

Julien Philip, Sébastien Morgenthaler, Michaël Gharbi, and George Drettakis. 2021. Free-viewpoint Indoor Neural Relighting from Multi-view Stereo. *ACM Transactions on Graphics* (2021). http://www-sop.inria.fr/reves/Basilic/2021/PMGD21

Trygve Randen and John Hakon Husoy. 1999. Filtering for texture classification: A comparative study. *IEEE Transactions on pattern analysis and machine intelligence* 21, 4 (1999), 291–310.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 2021. Vision Transformers for Dense Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12179–12188.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2022. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 3 (2022).

Constantino Carlos Reyes-Aldasoro and Abhir Bhalerao. 2006. The Bhattacharyya space for feature selection and its application to texture segmentation. *Pattern Recognition* 39, 5 (2006), 812–826.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

Gabriel Schwartz and Ko Nishino. 2013. Visual Material Traits: Recognizing Per-Pixel Material Context. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

Gabriel Schwartz and Ko Nishino. 2016. Material Recognition from Local Appearance in Global Context. *ArXiv* abs/1611.09394 (2016).

Gabriel Schwartz and Ko Nishino. 2019. Recognizing material properties from images. *IEEE transactions on pattern analysis and machine intelligence* 42, 8 (2019), 1981–1995.

Gabriel Schwartz and Ko Nishino. 2020. Recognizing Material Properties from Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 8 (2020), 1981–1995. https://doi.org/10.1109/TPAMI.2019.2907850

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

Maxim Tkachenko, Mikhail Malyuk, Nikita Shevchenko, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2021. Label Studio: Data labeling software. https://github.com/heartexlabs/label-studio Open source software available from https://github.com/heartexlabs/label-studio.

Sinisa Todorovic and Narendra Ahuja. 2009. Texel-based texture segmentation. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 841–848.

Paul Upchurch and Ransen Niu. 2022. A Dense Material Segmentation Dataset for Indoor and Outdoor Scene Parsing. In *European Conference on Computer Vision*. Springer, 450–466.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. 2021. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5463–5474.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. 2020b. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *European Conference on Computer Vision*. Springer, 108–126.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. 2020a. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems* 33 (2020), 17721–17732.

Jia Xue, Hang Zhang, Ko Nishino, and Kristin J. Dana. 2022. Differential Viewpoints for Ground Terrain Material Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 3 (2022), 1205–1218. https://doi.org/10.1109/TPAMI.2020.3025121

Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. 2020. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5791–5800.

Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. 2018. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916* (2018).

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. 2022. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11304–11314.

Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. 2019. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321* (2019).

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 633–641.

Xilong Zhou, Milos Hasan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Khademi Kalantari. 2022. TileGen: Tileable, Controllable Material Generation and Capture. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.