FC^4: Fully Convolutional Color Constancy with Confidence-weighted Pooling

Supplementary Material

Yuanming Hu[1*] Baoyuan Wang[2] Stephen Lin[2]
[1]Tsinghua University, [2]Microsoft Research
yuanmhu@gmail.com, {baoyuanw,stevelin}@microsoft.com

1. Analysis of Backpropagation

In Section 3.2 of the main paper, we provided a mathematical analysis of backpropagation in our network containing the novel weighted pooling layer. Here, we present this analysis in greater detail.

1.1. Prerequisites

Firstly, we derive the Jacobian $\mathbf{J}_{3\times3} = \frac{\partial\hat{\mathbf{x}}}{\partial\mathbf{x}}$ for a 3D vector $\mathbf{x}$. Each entry of $\mathbf{J}$ is deduced as follows:

$$\begin{aligned}
\mathbf{J}_{\mathrm{ij}} &= \left[\partial\left(\frac{\mathbf{x}}{\|\mathbf{x}\|_2}\right)\Big/\partial\mathbf{x}\right]_{\mathrm{ij}} && (1)\\
&= \partial\left(\frac{\mathbf{x}_\mathrm{i}}{\|\mathbf{x}\|_2}\right)\Big/\partial\mathbf{x}_\mathrm{j} && (2)\\
&= \frac{1}{\|\mathbf{x}\|_2^2}\left(\|\mathbf{x}\|_2\frac{\partial\mathbf{x}_\mathrm{i}}{\partial\mathbf{x}_\mathrm{j}} - \mathbf{x}_\mathrm{i}\frac{\partial\|\mathbf{x}\|_2}{\partial\mathbf{x}_\mathrm{j}}\right) && (3)\\
&= \frac{1}{\|\mathbf{x}\|_2}\left(\frac{\partial\mathbf{x}_\mathrm{i}}{\partial\mathbf{x}_\mathrm{j}} - \hat{\mathbf{x}}_\mathrm{i}\frac{\partial\|\mathbf{x}\|_2}{\partial\mathbf{x}_\mathrm{j}}\right) && (4)\\
&= \frac{1}{\|\mathbf{x}\|_2}\left(\delta_{\mathrm{ij}} - \hat{\mathbf{x}}_\mathrm{i}\frac{\partial\left(\sum_\mathrm{k}\mathbf{x}_\mathrm{k}^2\right)^{\frac{1}{2}}}{\partial\mathbf{x}_\mathrm{j}}\right) && (5)\\
&= \frac{1}{\|\mathbf{x}\|_2}\left(\delta_{\mathrm{ij}} - \hat{\mathbf{x}}_\mathrm{i}\frac{\mathbf{x}_\mathrm{j}}{\|\mathbf{x}\|_2}\right) && (6)\\
&= \frac{1}{\|\mathbf{x}\|_2}\left(\delta_{\mathrm{ij}} - \hat{\mathbf{x}}_\mathrm{i}\hat{\mathbf{x}}_\mathrm{j}\right), && (7)
\end{aligned}$$

where

$$\delta_{\mathrm{ij}} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}. \qquad (8)$$

In matrix notation,

$$\mathbf{J} = \frac{1}{\|\mathbf{x}\|_2}\left(\mathbf{I}_3 - \hat{\mathbf{x}}\otimes\hat{\mathbf{x}}\right) = \frac{1}{\|\mathbf{x}\|_2}\mathbf{\Theta}_{\hat{\mathbf{x}}}, \qquad (9)$$

*This work was done when Yuanming Hu was an intern at Microsoft Research.

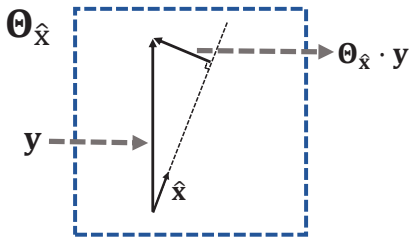


Figure 1: An illustration of $\mathbf{\Theta}_{\hat{\mathbf{x}}}$.

where $\mathbf{I}_3$ is the 3×3 identity matrix and "$\otimes$" denotes tensor product. $\mathbf{\Theta}_{\hat{\mathbf{x}}} = \mathbf{I}_3 - \hat{\mathbf{x}}\otimes\hat{\mathbf{x}}$ is a **symmetric** matrix that takes as input a vector $\mathbf{y}$ and outputs the orthogonal part of $\mathbf{y}$ with respect to $\hat{\mathbf{x}}$, as illustrated in Figure 1. Substituting $\mathbf{x}$ with $\mathbf{p}_\mathrm{g}$, we have

$$\frac{\partial\hat{\mathbf{p}}_\mathrm{g}}{\partial\mathbf{p}_\mathrm{g}} = \frac{1}{\|\mathbf{p}_\mathrm{g}\|_2}\mathbf{\Theta}_{\hat{\mathbf{p}}_\mathrm{g}}. \qquad (10)$$

Secondly, to facilitate differentiation, we extend the domain of the angular loss function L from $\{\mathbf{x}|\mathbf{x}\in\mathcal{R}_+^3, \|\mathbf{x}\|_2 = 1\}$ to $\{\mathbf{x}|\mathbf{x}\in\mathcal{R}_+^3\}$, dropping the normalization constraint. We then define the extended loss function L as

$$L(\mathbf{x}) = \arccos\left(\frac{\mathbf{x}}{\|\mathbf{x}\|_2}\cdot\hat{\mathbf{p}}_\mathrm{g}^*\right). \qquad (11)$$

We point out one important property of this loss function – that its gradient with respect to $\mathbf{x}$ is orthogonal to $\mathbf{x}$, or

$$\mathbf{x}^T\left(\frac{\partial L(\mathbf{x})}{\partial\mathbf{x}}\right)^T = 0, \qquad (12)$$

since increasing the length of $\mathbf{x}$ does not affect the angular loss, which is defined purely on the direction of $\mathbf{x}$, *i.e.* $\mathbf{x}/\|\mathbf{x}\|_2$.

From its symmetry and the fact that $\mathbf{\Theta}_{\hat{\mathbf{p}}_\mathrm{g}}$ returns the orthogonal component of a vector with respect to $\hat{\mathbf{p}}_\mathrm{g}$, we have

$$\frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}} \tag{13}$$

$$= \left(\mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}}^{T} \cdot \left(\frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \right)^{T} \right)^{T} \tag{14}$$

$$= \left(\mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}} \cdot \left(\frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \right)^{T} \right)^{T} \tag{15}$$

$$= \left(\left(\frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \right)^{T} \right)^{T} \tag{16}$$

$$= \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}}. \tag{17}$$

As we can see, the term $\mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}}$ is eliminated.

1.2. Backpropagation

With the aforementioned prerequisites, we have all we need to derive and simplify the derivative of the loss function, with respect to each local estimate $\hat{\mathbf{p}}_{\mathrm{i}}$ and confidence c_{i}:

$$\frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{i}}} = \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \frac{\partial \hat{\mathbf{p}}_{\mathrm{g}}}{\partial \mathbf{p}_{\mathrm{g}}} \cdot \frac{\partial \mathbf{p}_{\mathrm{g}}}{\partial \hat{\mathbf{p}}_{\mathrm{i}}} \tag{18}$$

$$= \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \frac{1}{\|\mathbf{p}_{\mathrm{g}}\|_2} \mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}} \cdot c_{\mathrm{i}} \mathbf{I}_3 \tag{19}$$

$$= \frac{c_{\mathrm{i}}}{\|\mathbf{p}_{\mathrm{g}}\|_2} \cdot \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}} \tag{20}$$

$$= \frac{c_{\mathrm{i}}}{\|\mathbf{p}_{\mathrm{g}}\|_2} \cdot \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \tag{21}$$

and

$$\frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial c_{\mathrm{i}}} = \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \frac{\partial \hat{\mathbf{p}}_{\mathrm{g}}}{\partial \mathbf{p}_{\mathrm{g}}} \cdot \frac{\partial \mathbf{p}_{\mathrm{g}}}{\partial c_{\mathrm{i}}} \tag{22}$$

$$= \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \frac{1}{\|\mathbf{p}_{\mathrm{g}}\|_2} \mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}} \cdot \hat{\mathbf{p}}_{\mathrm{i}} \tag{23}$$

$$= \frac{1}{\|\mathbf{p}_{\mathrm{g}}\|_2} \cdot \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \mathbf{\Theta}_{\hat{\mathbf{p}}_{\mathrm{g}}} \cdot \hat{\mathbf{p}}_{\mathrm{i}} \tag{24}$$

$$= \frac{1}{\|\mathbf{p}_{\mathrm{g}}\|_2} \cdot \frac{\partial L(\hat{\mathbf{p}}_{\mathrm{g}})}{\partial \hat{\mathbf{p}}_{\mathrm{g}}} \cdot \hat{\mathbf{p}}_{\mathrm{i}}. \tag{25}$$

1.3. Discussions

There are two main takeaways from this analysis of the backpropagation. The first is that the strength of the supervision signal toward a local estimate is proportional to the confidence for its local area, as we can see from Equation 21. Notice in the equation that for all of the local estimates there is the same global gradient direction, and that they differ only in magnitude according to confidence c_{i}. Since the supervision focuses on local areas with higher confidence, the network essentially concentrates its learning on areas helpful for estimation, while disregarding "noisy" regions of low confidence.

The second takeaway is that, as seen from Equation 25, the supervision for confidence values depends on whether a local estimate lies along a direction that leads to a better global estimate. If a local estimate is helpful in improving the global estimate, then its confidence will increase. Otherwise, it is reduced. In this way, the network learns how to pool local estimates to produce a desired global result.

The entire training cycle of our method is illustrated in Figure 2.

2. Output Visualizations

In this section, we show some outputs generated on both training and test images in the two datasets. We found that most ground truth tends to be green, possibly because of the high sensitivity of the green filter in many cameras. Though aggressive data augmentation is done for training, some of the augmented image crops are parts of the original full images; therefore, we directly visualize results on full training images, with dropout disabled. Generally we find that confidence maps on training images are slightly sharper compared with those on test images, likely due to some degree of overfitting that inevitably exists in supervised training. We use three-fold cross-validation, so when two identical images appear in both training and test, they are from different folds.

For AlexNet-FC[4], on the reprocessed Color Checker Dataset, training results are shown in Figure 4 while test results are shown in Figure 5, 6 and 7. In Figure 4 and Figure 5 we show the same set of images from different folds, where they are used for training and testing respectively. On the NUS 8-Camera Dataset, training and test results on a same set of images (but from different folds) are shown in Figure 8 and Figure 9. In addition, we show test results of SqueezeNet-FC[4] on the reprocessed Color Checker Dataset in Figure 10.

3. Average Confidence Map Value

As discussed in the end of the main paper, the average value of a learned confidence map can serve as the "global confidence" of our method. This property is verified by the fact that higher confidence indicates lower estimation error, as shown in Figure 3. In conclusion, confidence values are meaningful not only within a single image, but also across different images.

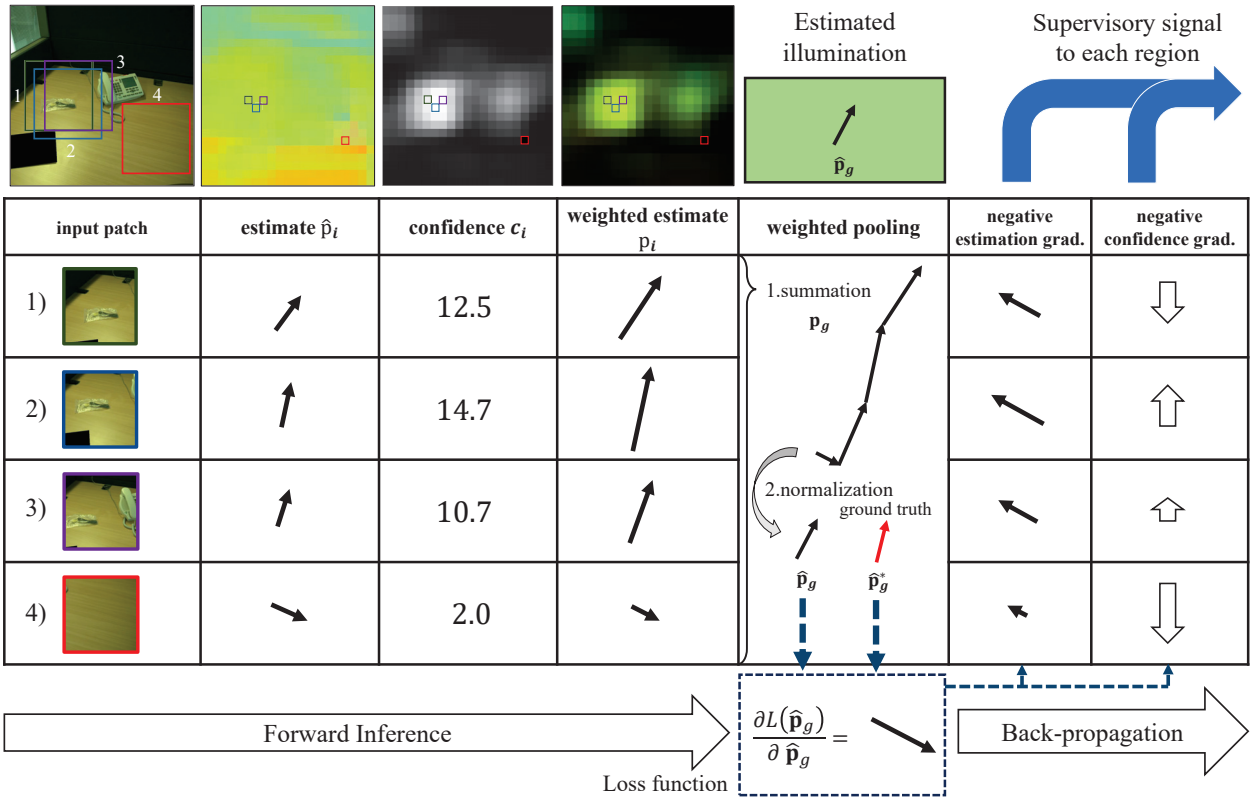


Figure 2: An overview of the training cycle. We select four representative regions and provide a 2D illustration for the original 3D (RGB) vectors. Note that for "negative confidence gradient", we use upward/downward arrows for increasing/decreasing each corresponding confidence, which is a scalar value.

Figure 3: Average estimation error vs. average confidence map value. For SqueezeNet-FC^4 on the reprocessed Color Checker dataset, we sort the images with respect to the average confidence map value and group them into five equally-sized bins. The plot indicates that higher confidence is associated with lower average error.

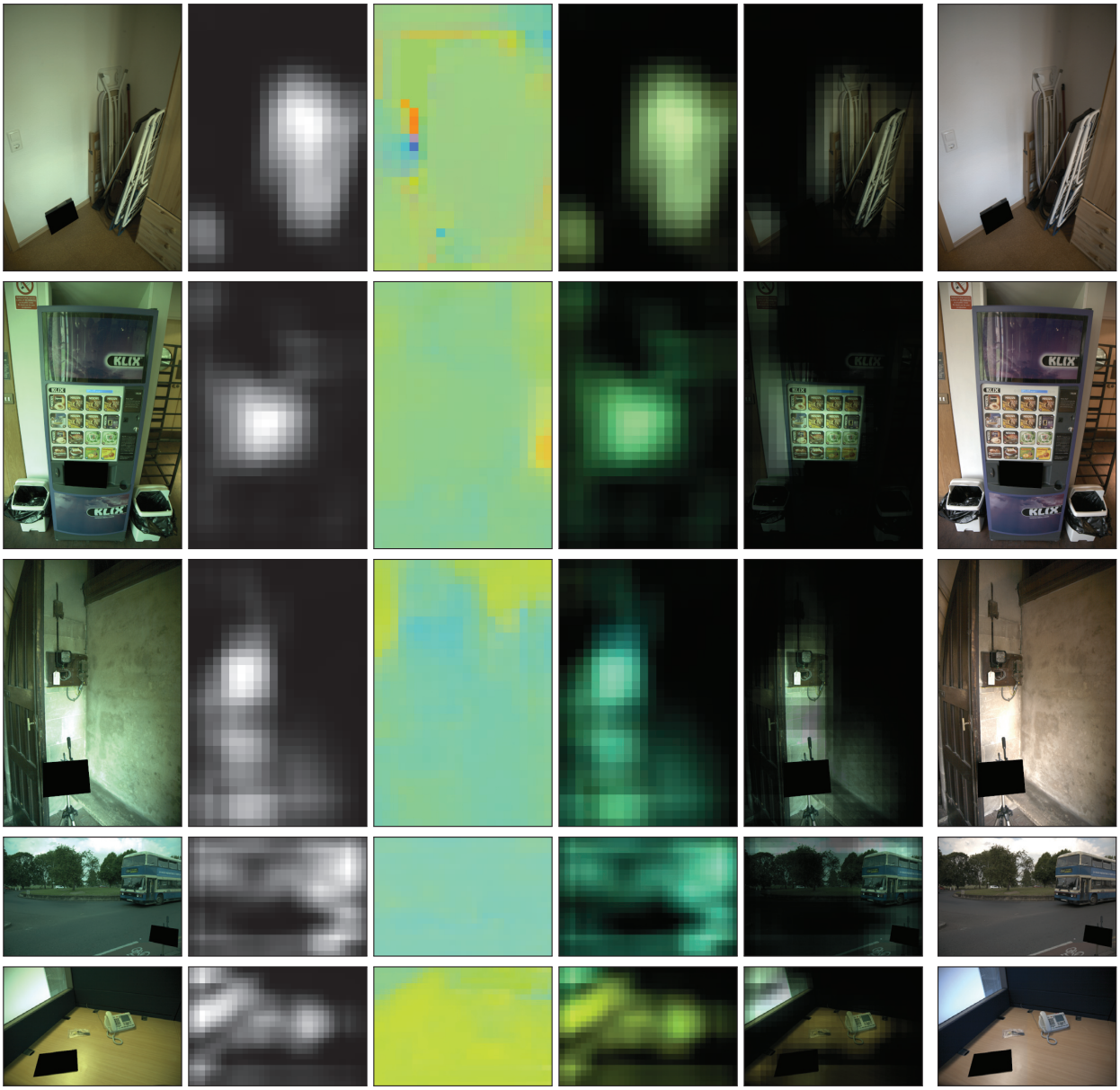


Figure 4: Examples of AlexNet-FC^4 **training** outputs by the network on the reprocessed Color Checker Dataset. Note that ambiguous regions of little semantic value are masked by the confidence map, so that the network is protected from learning these noisy labels.

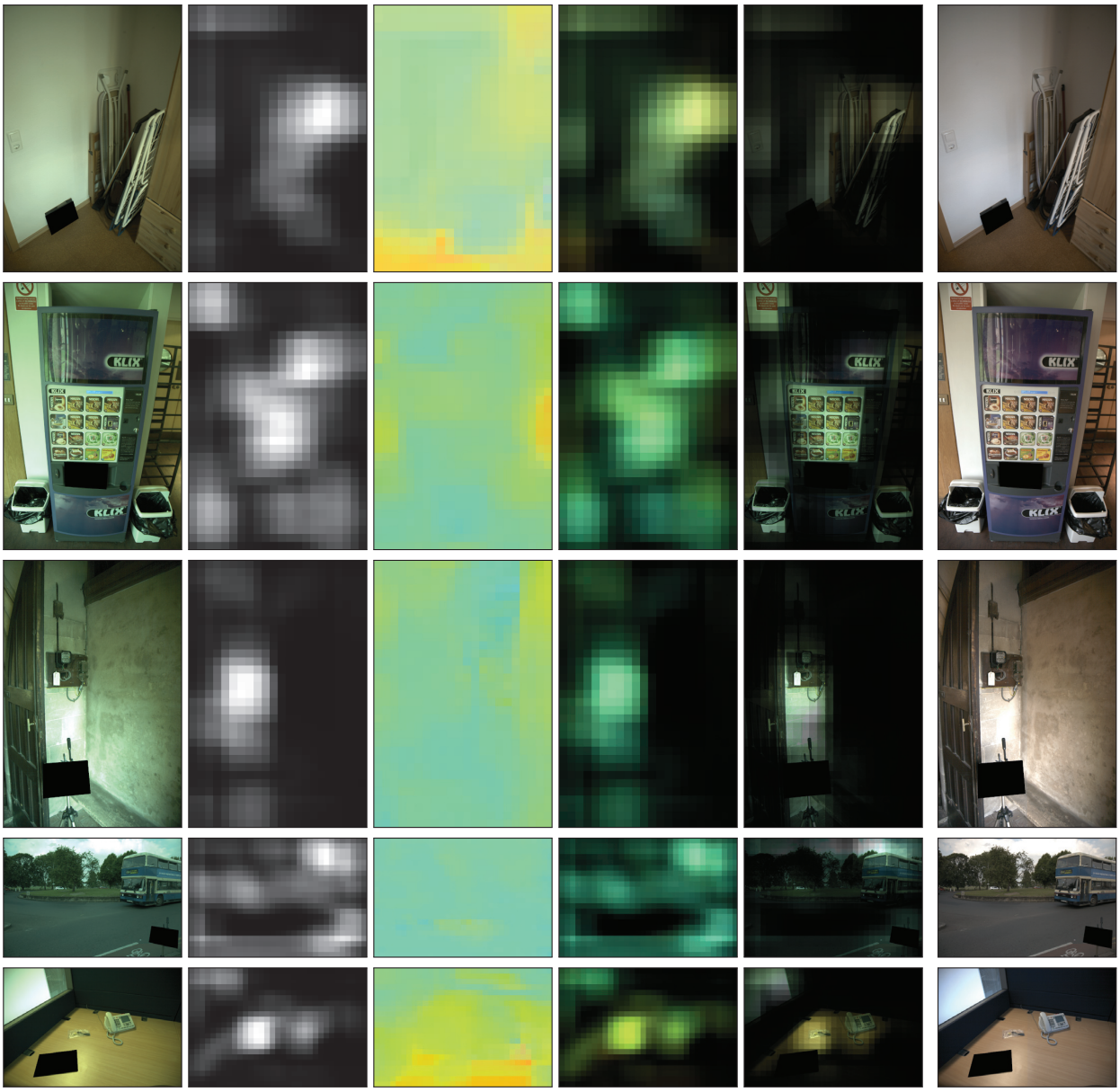


Figure 5: Examples of AlexNet-FC^4 **test** outputs by the network on the reprocessed Color Checker Dataset. Note that noisy estimates in regions of little semantic value are masked by the confidence map, resulting in more robust estimation.

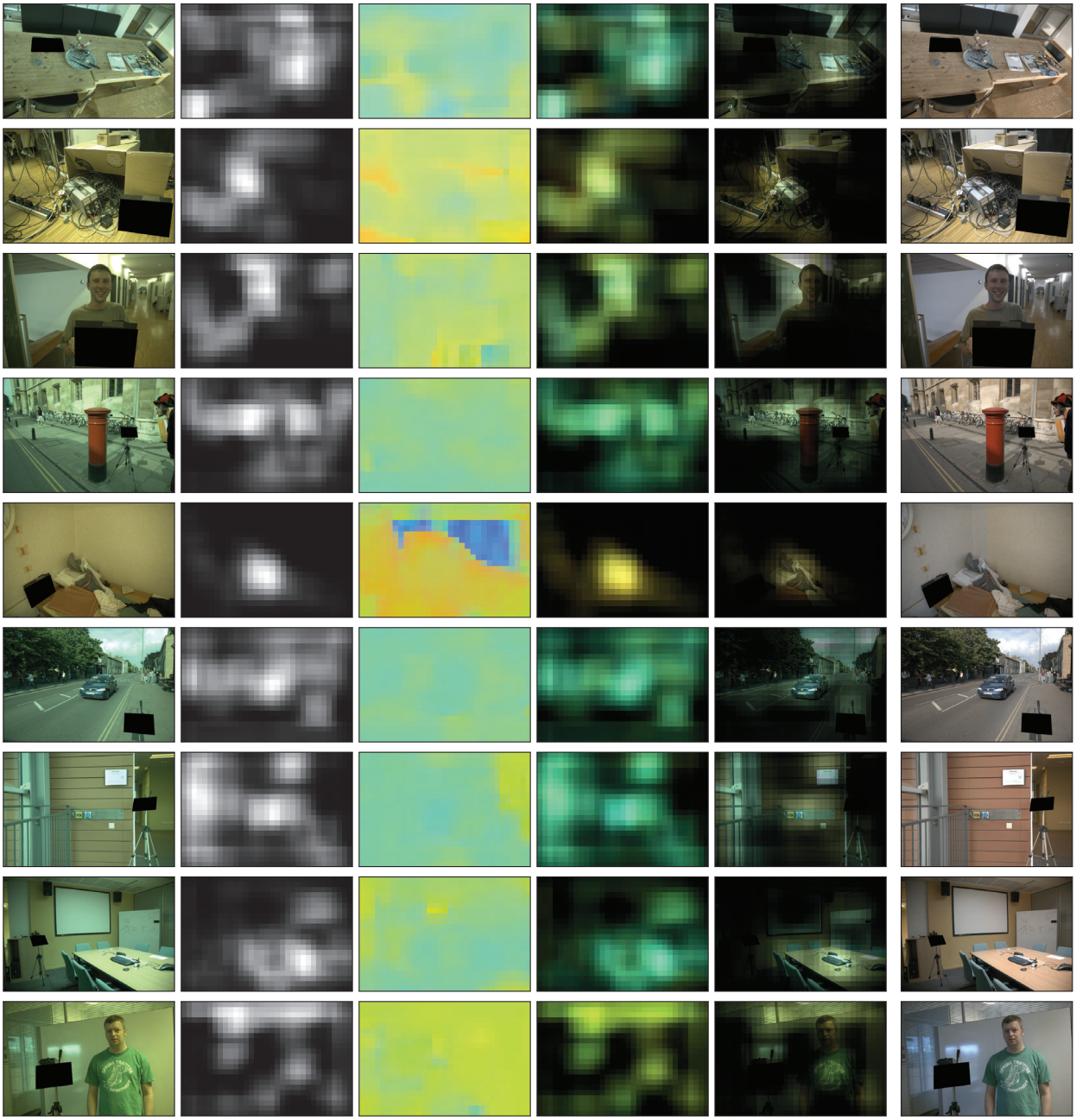


Figure 6: More AlexNet-FC^4 **test** output examples on the reprocessed Color Checker Dataset.

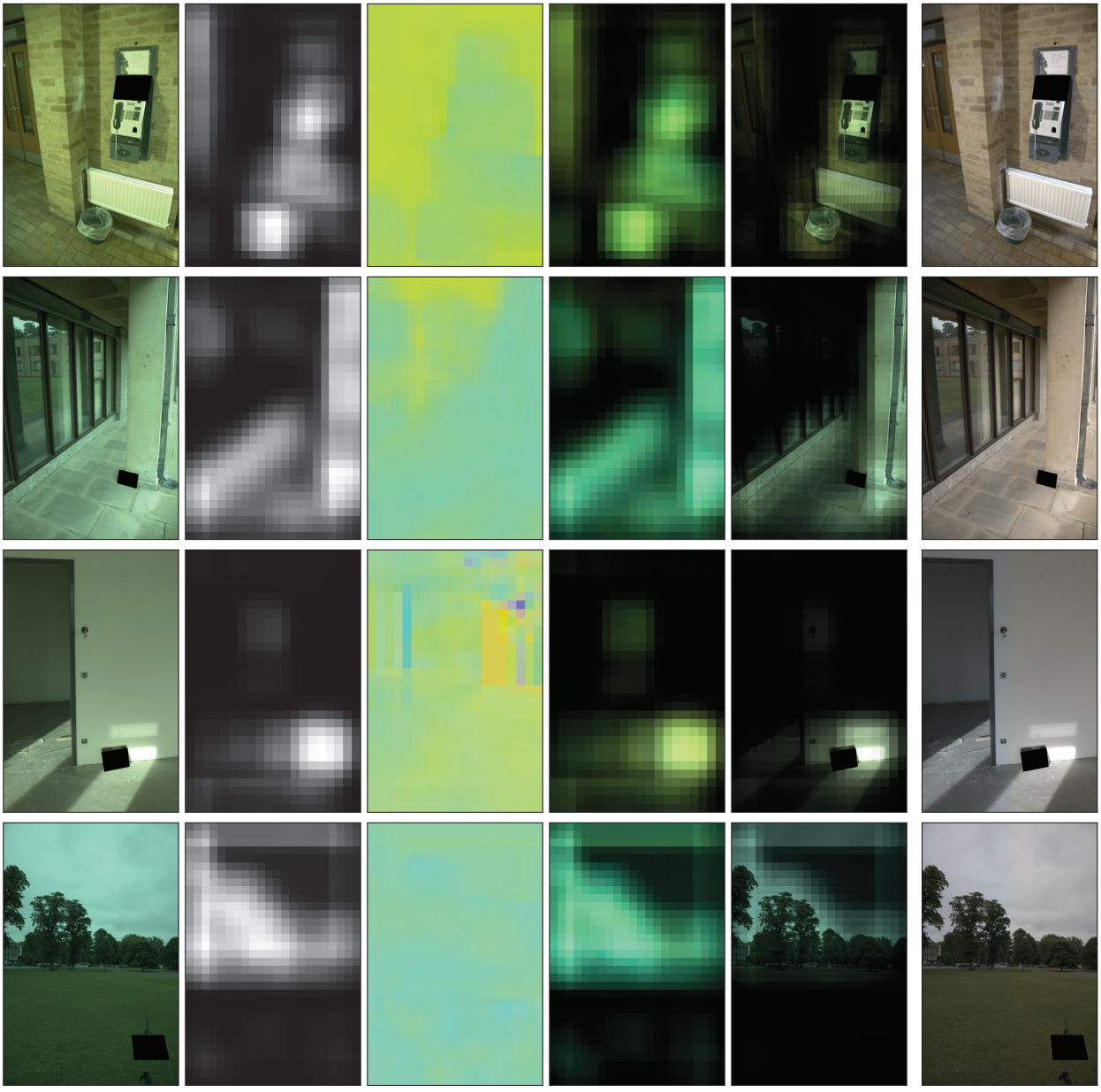
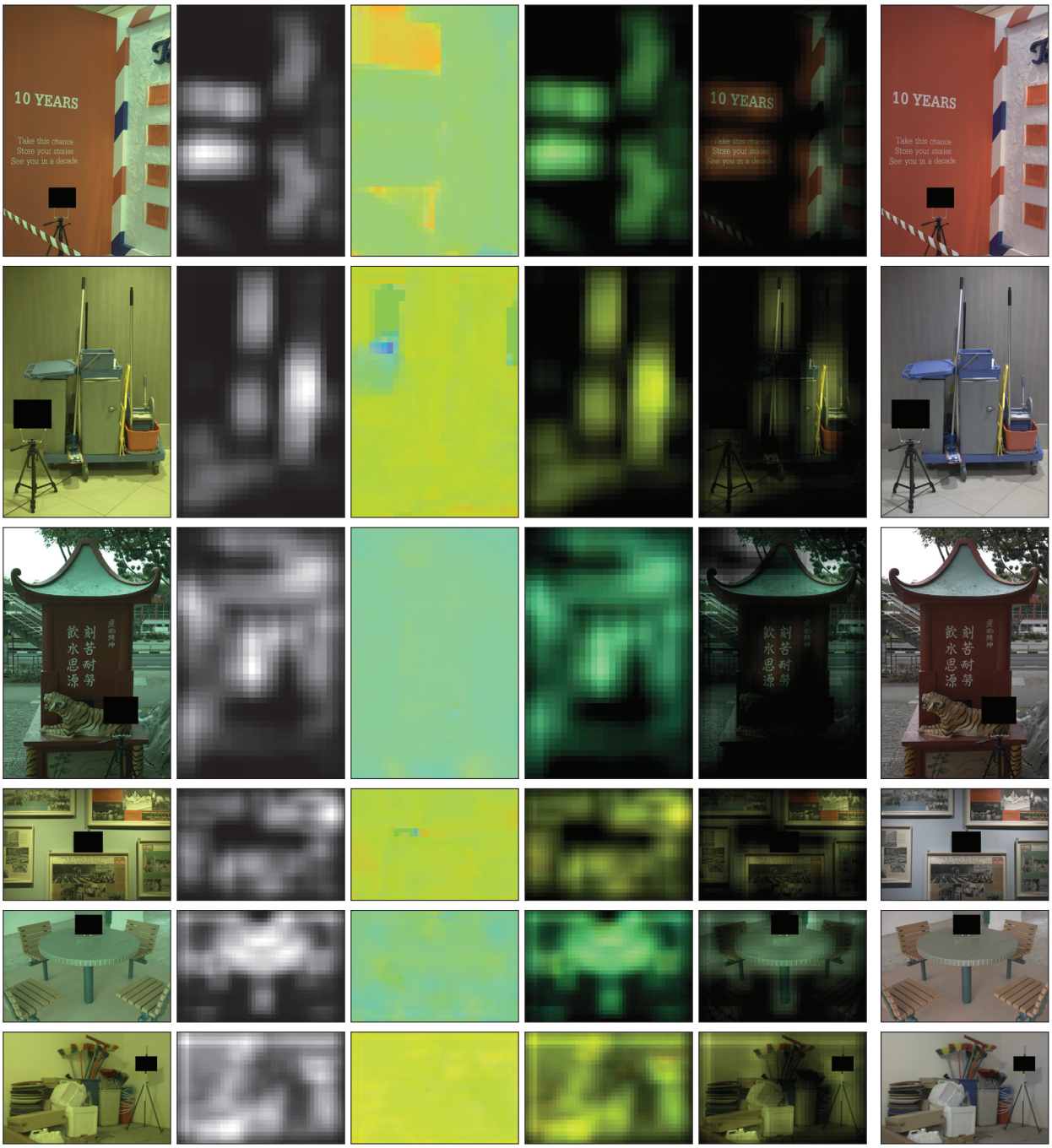
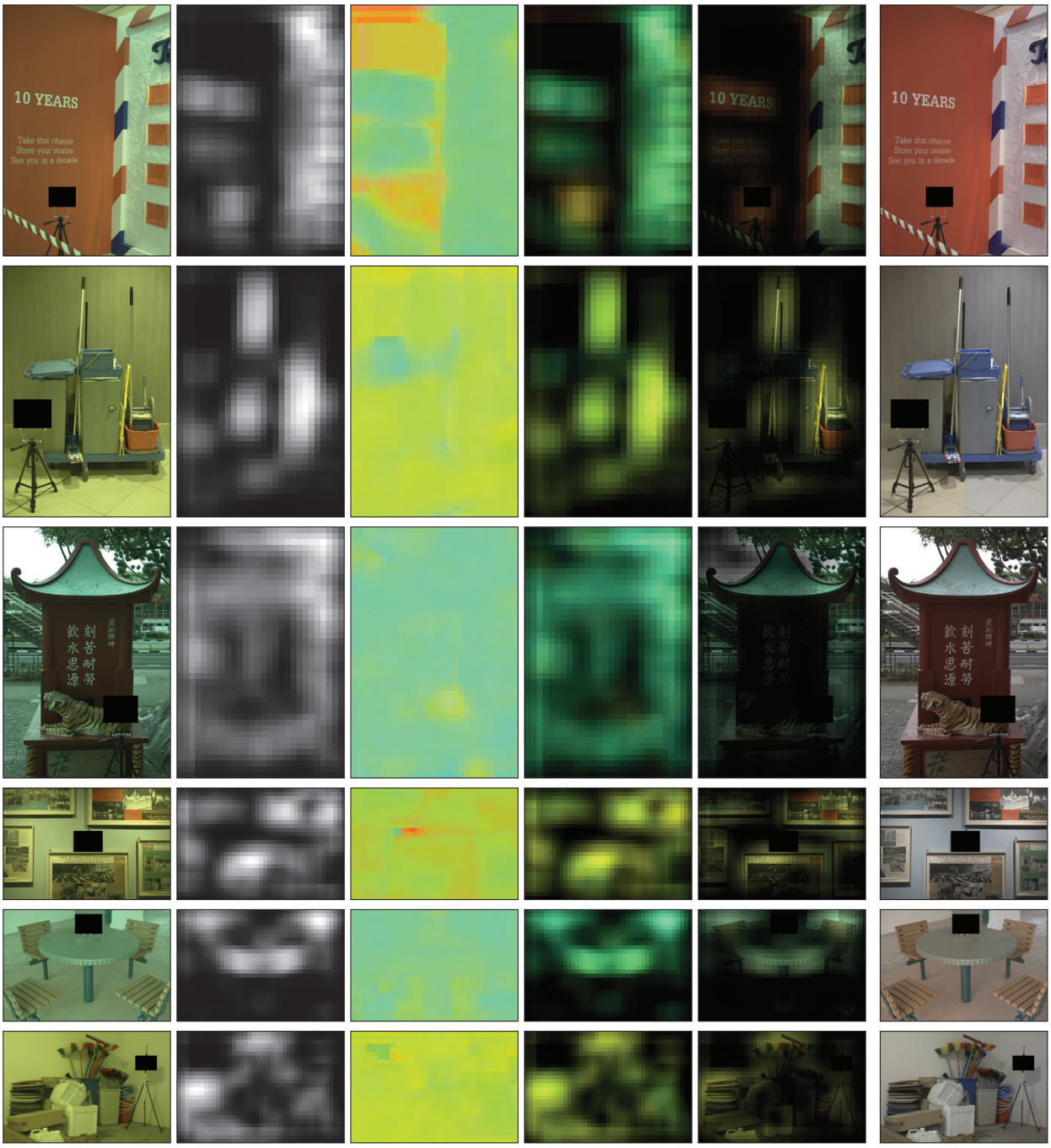


Figure 7: More AlexNet-FC^4 **test** output examples on the reprocessed Color Checker Dataset.



input image | confidence map | estimation map | weighted estimation | image × confidence | estimation (left half) and ground truth (right half)

Figure 8: Examples of AlexNet-FC^4 **training** outputs by the network on the NUS 8-Camera Dataset. Note that ambiguous regions of little semantic value are masked by the confidence map, so that the network is protected from learning these noisy labels.



input image | confidence map | estimation map | weighted estimation | image × confidence | estimation (left half) and ground truth (right half)

Figure 9: Examples of AlexNet-FC^4 **test** outputs on the NUS 8-Camera Dataset.

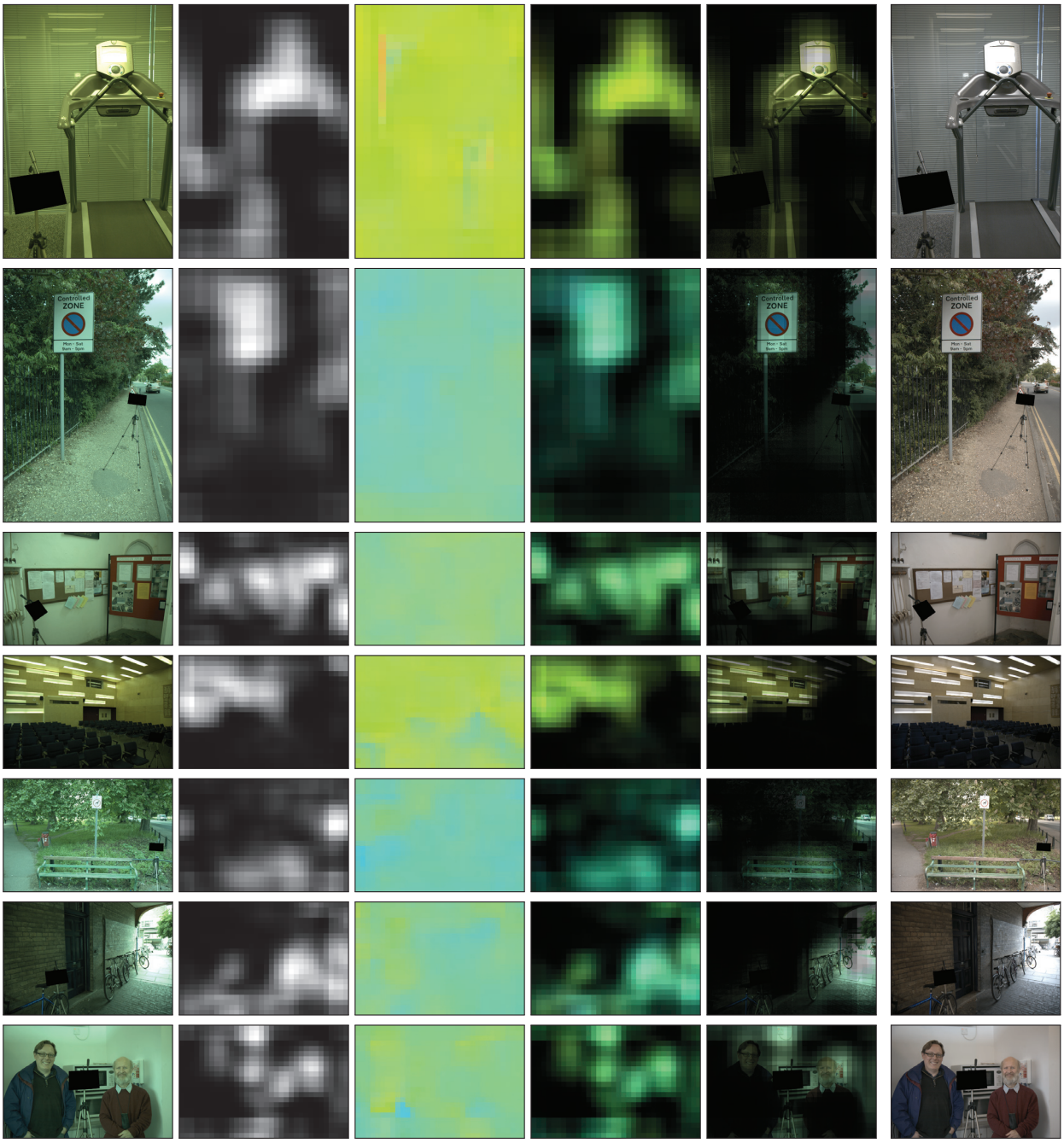


Figure 10: Examples of SqueezeNet-FC^4 **test** outputs on the reprocessed Color Checker Dataset.