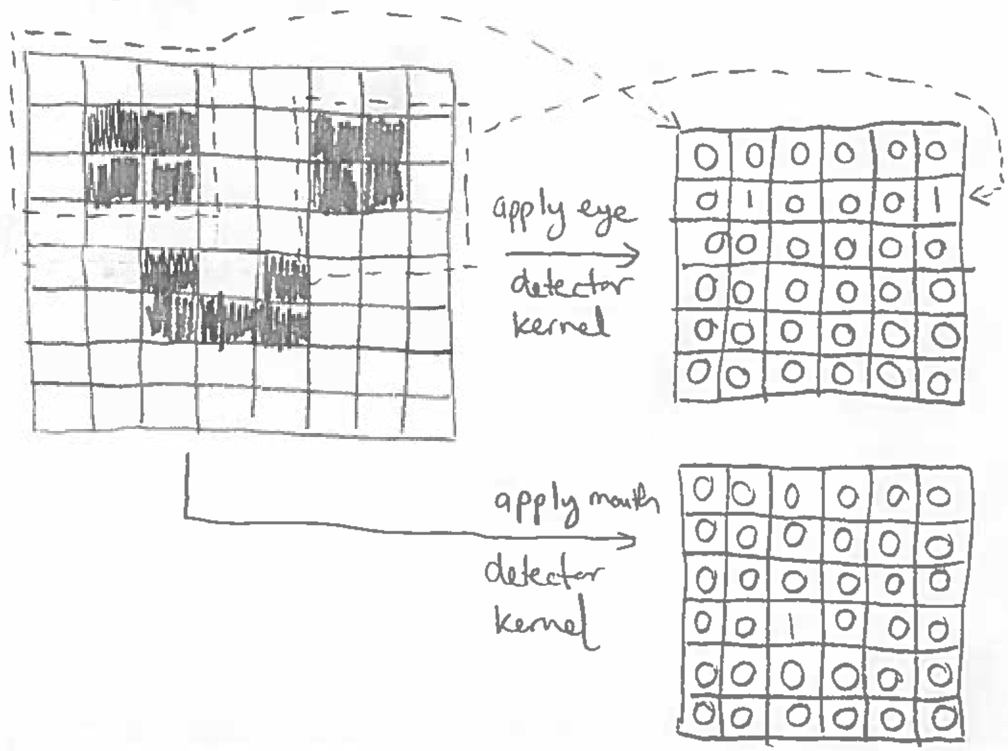
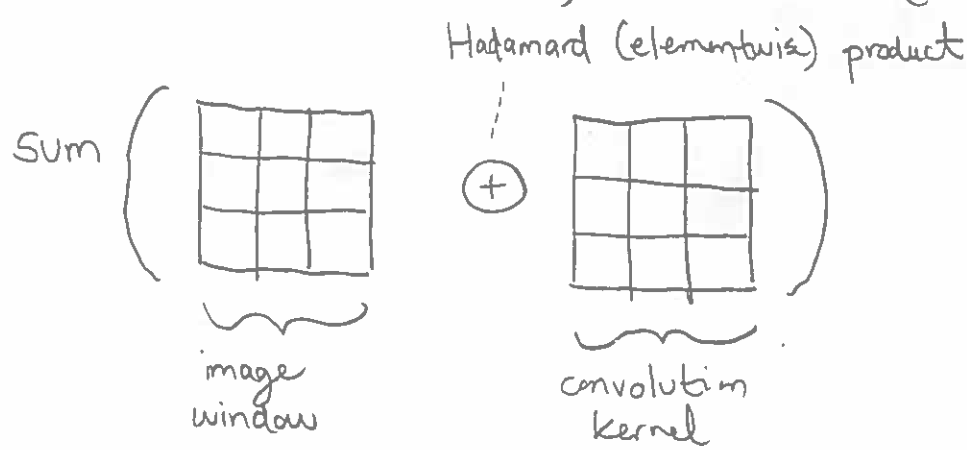
Computing Convolutions

① When we run the happy face detector, we need to apply each kernel to each 3×3 window of the image:



② By "apply a kernel to a window," we mean that we compute:



Computing Convolutions

③ How can we do this efficiently (i.e. leverage matrix multiplication libraries as much as possible)?

Let's consider a simpler example, in which we have a 3×3 "image":

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$$

and a single 2×2 kernel:

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}$$

---

④ To "convolve" the image with the kernel (with no padding and stride 1), we want to compute the matrix:

$$\begin{array}{|c|c|}\hline \text{sum}\left(\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}\right) & \text{sum}\left(\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}\right) \\ \hline \text{sum}\left(\begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}\right) & \text{sum}\left(\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}\right) \\ \hline \end{array}$$

$$= \begin{array}{|c|c|}\hline 3 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$$

Computing Convolutions

⑤ We can express these computations as a matrix multiplication by expressing the kernel as a row vector:

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline\end{array} \longrightarrow \begin{bmatrix} 1 & 2 & 0 & -1 \end{bmatrix}$$

and the image windows as columns of a matrix:

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline\end{array} \longrightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

⑥ When we mulbply the row vector by the column matrix, we get:

$$\begin{bmatrix} 1 & 2 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{sum}\left(\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline\end{array} \oplus \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline\end{array} \right)$$

which we can reshape into a 2x2 matrix to get the desired result:

$$\begin{bmatrix} 3 & 0 & 0 & 1 \end{bmatrix} \longrightarrow \begin{array}{|c|c|}\hline 3 & 0 \\ \hline 0 & 1 \\ \hline\end{array}$$

Computing Convolutions

⑦ It's not too hard to do this with multiple kernels at once (e.g. an eye detector and a mouth detector). Let's add a kernel to our example, so we have now a 3x3 image: $\begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$

and two 2x2 kernels: $\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}$ and $\begin{array}{|c|c|}\hline 3 & 0 \\ \hline -2 & 4 \\ \hline \end{array}$

⑧ Convolving the image with these kernels (with no padding and stride 1), we get:

$$\left[\begin{array}{|c|c|}\hline \text{Sum}\left(\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 1&2\\\hline 0&-1\\\hline\end{array}\right) & \text{Sum}\left(\begin{array}{|c|c|}\hline 1&0\\\hline 0&1\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 1&2\\\hline 0&-1\\\hline\end{array}\right) \\ \hline \text{Sum}\left(\begin{array}{|c|c|}\hline 0&0\\\hline 1&0\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 1&2\\\hline 0&-1\\\hline\end{array}\right) & \text{Sum}\left(\begin{array}{|c|c|}\hline 0&1\\\hline 0&1\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 1&2\\\hline 0&-1\\\hline\end{array}\right) \\ \hline \end{array}\ ,\ \begin{array}{|c|c|}\hline \text{Sum}\left(\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 3&0\\\hline -2&4\\\hline\end{array}\right) & \text{Sum}\left(\begin{array}{|c|c|}\hline 1&0\\\hline 0&1\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 3&0\\\hline -2&4\\\hline\end{array}\right) \\ \hline \text{Sum}\left(\begin{array}{|c|c|}\hline 0&0\\\hline 1&0\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 3&0\\\hline -2&4\\\hline\end{array}\right) & \text{Sum}\left(\begin{array}{|c|c|}\hline 0&1\\\hline 0&1\\\hline\end{array} \oplus \begin{array}{|c|c|}\hline 3&0\\\hline -2&4\\\hline\end{array}\right) \\ \hline \end{array}\right]$$

$$= \left[\begin{array}{|c|c|}\hline 3 & 0 \\ \hline 0 & 1 \\ \hline \end{array}\ ,\ \begin{array}{|c|c|}\hline 3 & 7 \\ \hline -2 & -2 \\ \hline \end{array}\right]$$

Computing Convolutions

(9) We can perform the same trick, expressing each kernel as a row in a "kernel-row" matrix, and each image window as a column in a "window column" matrix. Then we multiply the matrices to obtain our desired quantities:

$$\begin{array}{l} \text{kernel 1} \rightarrow \\ \text{kernel 2} \rightarrow \end{array} \underbrace{\begin{bmatrix} 1 & 2 & 0 & -1 \\ 3 & 0 & -2 & 4 \end{bmatrix}}_{\text{kernel-row matrix}} \overbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}^{\text{window-column matrix}} = \begin{bmatrix} 3 & 0 & 0 & 1 \\ 3 & 7 & -2 & -2 \end{bmatrix}$$

$$\begin{array}{cccc} \uparrow & & & \uparrow \\ \text{image window 1} & & & \text{image window 4} \end{array}$$

which we can reshape into a $2 \times 2 \times 2$ tensor to get the desired result:

$$\begin{bmatrix} 3 & 0 & 0 & 1 \\ 3 & 7 & -2 & -2 \end{bmatrix} \rightarrow \left[\begin{array}{|c|c|} \hline 3 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \, , \, \begin{array}{|c|c|} \hline 3 & 7 \\ \hline -2 & -2 \\ \hline \end{array} \right]$$

Computing Convolutions

(10) There are a couple additional complications. While grayscale images are stored as matrices, color images are often represented by order-3 tensors — one matrix for each red, green, and blue "channel", e.g.

$$\left[\underset{\text{red channel}}{\begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix}},\ \underset{\text{green channel}}{\begin{bmatrix}1&1&0\\0&0&1\\1&0&1\end{bmatrix}},\ \underset{\text{blue channel}}{\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}}\right]$$

(11) In this case, each kernel is also an order-3 tensor, one matrix for each channel, e.g:

$$\left[\underset{\text{red kernel}}{\begin{bmatrix}1&0\\0&0\end{bmatrix}},\ \underset{\text{green kernel}}{\begin{bmatrix}0&0\\1&0\end{bmatrix}},\ \underset{\text{blue kernel}}{\begin{bmatrix}-1&0\\0&0\end{bmatrix}}\right]$$

(12) The convolution is:

$\text{sum}\left(\begin{bmatrix}1&0\\0&1\end{bmatrix}\oplus\begin{bmatrix}1&0\\0&0\end{bmatrix}\right)+\text{sum}\left(\begin{bmatrix}1&1\\0&0\end{bmatrix}\oplus\begin{bmatrix}0&0\\1&0\end{bmatrix}\right)+\text{sum}\left(\begin{bmatrix}0&0\\0&1\end{bmatrix}\oplus\begin{bmatrix}-1&0\\0&0\end{bmatrix}\right)$	$\text{sum}\left(\begin{bmatrix}0&1\\1&0\end{bmatrix}\oplus\begin{bmatrix}1&0\\0&0\end{bmatrix}\right)+\text{sum}\left(\begin{bmatrix}1&0\\0&1\end{bmatrix}\oplus\begin{bmatrix}0&0\\1&0\end{bmatrix}\right)+\text{sum}\left(\begin{bmatrix}0&1\\1&0\end{bmatrix}\oplus\begin{bmatrix}-1&0\\0&0\end{bmatrix}\right)$

$$= \begin{bmatrix}1&0\\1&0\end{bmatrix}$$

Computing Convolutions

(13) Ok, no problem! We can still compute this as the multiplication of a kernel-row matrix with a window-column matrix: we just need to concatenate the channels.

$$
\begin{array}{c c}
 & \begin{array}{ccc} \text{red} & \text{green} & \text{blue} \end{array} \\
\text{kernel 1} \rightarrow & \left[\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \right]
\end{array}
\cdot
\begin{array}{c c}
\begin{array}{cccc} \text{window1} & \text{window2} & \text{window3} & \text{window4} \end{array} & \\
\left[\begin{array}{cccc}
\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \\
\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}
\end{array} \right] &
\begin{array}{l} \leftarrow \text{red} \\ \\ \leftarrow \text{green} \\ \\ \leftarrow \text{blue} \end{array}
\end{array}
$$

$$
= \text{kernel 1} \rightarrow \left[\underbrace{1\ 0\ 0\ 0}_{\text{red}}\ \underbrace{0\ 0\ 1\ 0}_{\text{green}}\ \underbrace{-1\ 0\ 0\ 0}_{\text{blue}} \right]
\begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\begin{array}{l} \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} \text{red} \\ \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} \text{green} \\ \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} \text{blue} \end{array}
$$

$$
\begin{array}{cccc} \uparrow & \uparrow & \uparrow & \uparrow \\ \text{window1} & \text{window2} & \text{window3} & \text{window4} \end{array}
$$

$$
= \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}
$$

Computing Convolutions

(14) The resulting matrix can then be reshaped into a $\underset{\substack{\uparrow \\ \text{num} \\ \text{kernels}}}{1} \times \underbrace{2 \times 2}_{\substack{\uparrow \\ \text{resulting image} \\ \text{size}}}$ tensor to get the result of convolution:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \end{bmatrix}$$

(15) All right. That's almost, but not quite, everything. The last complication is, in minibatch training, we're usually not convolving one image at a time, but actually a batch of images.

So the input to our convolution might actually be an order-4 tensor like this:

$$\begin{array}{c c}
\begin{bmatrix}
\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\
\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}
\end{bmatrix} & \begin{array}{l} \leftarrow \text{image } 1 \\[3em] \leftarrow \text{image } 2 \end{array} \\
\begin{array}{ccc} \quad\uparrow\quad & \quad\uparrow\quad & \quad\uparrow\quad \\ \text{red} & \text{green} & \text{blue} \end{array} &
\end{array}$$

This tensor has shape $\overset{\text{num images}}{2} \times \underset{\uparrow\,\text{RGB}}{3} \times \overbrace{3 \times 3}^{\text{image size}}$.

Computing Convolutions

(16) This is a straightforward (ish) extension. The kernels haven't changed, so the kernel-row matrix doesn't change. We just need to add the other image's windows as additional columns of the window-column matrix.

$$
\begin{array}{r}
\\ \text{kernel 1} \rightarrow
\end{array}
\begin{array}{ccc}
\text{red} & \text{green} & \text{blue} \\
\downarrow & \downarrow & \downarrow
\end{array}
$$

$$
\text{kernel 1} \rightarrow
\left[\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline\end{array} \;\; \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \;\; \begin{array}{|c|c|}\hline -1 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \right]
\begin{array}{c}
\overbrace{\begin{array}{cccc} \text{win 1} & \text{win 2} & \text{win 3} & \text{win 4} \end{array}}^{\text{image 1}}
\overbrace{\begin{array}{cccc} \text{win 1} & \text{win 2} & \text{win 3} & \text{win 4} \end{array}}^{\text{image 2}} \\
\left[\begin{array}{cccccccc}
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline\end{array} \\
\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \\
\begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 1 \\ \hline\end{array} &
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 1 \\ \hline\end{array}
\end{array} \right]
\end{array}
\begin{array}{l}
\\ \leftarrow \text{red} \\ \leftarrow \text{green} \\ \leftarrow \text{blue}
\end{array}
$$

$$
= \left[\underbrace{1\;0\;0\;1}_{\text{red}} \;\; \underbrace{0\;0\;1\;0}_{\text{green}} \;\; \underbrace{-1\;0\;0\;0}_{\text{blue}} \right]
\begin{array}{c}
\overbrace{\begin{array}{cccc} \text{win 1} & \text{win 2} & \text{win 3} & \text{win 4} \end{array}}^{\text{image 1}}
\overbrace{\begin{array}{cccc} \text{win 1} & \text{win 2} & \text{win 3} & \text{win 4} \end{array}}^{\text{image 2}} \\
\left[\begin{array}{cccccccc}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{array} \right]
\end{array}
\begin{array}{l}
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \text{red} \\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \text{green} \\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \text{blue}
\end{array}
$$

$$
= \left[1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 2 \;\; 2 \;\; 1 \right]
$$

which can be reshaped into a $\underset{\text{num images}}{2} \times \underset{\text{num kernels}}{1} \times \overbrace{2 \times 2}^{\text{image size}}$ tensor to get the result of convolution:

$$
\left[\begin{array}{c}
\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \\[1em]
\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 2 & 1 \\ \hline\end{array}
\end{array} \right]
$$