

# Convolution & Pooling

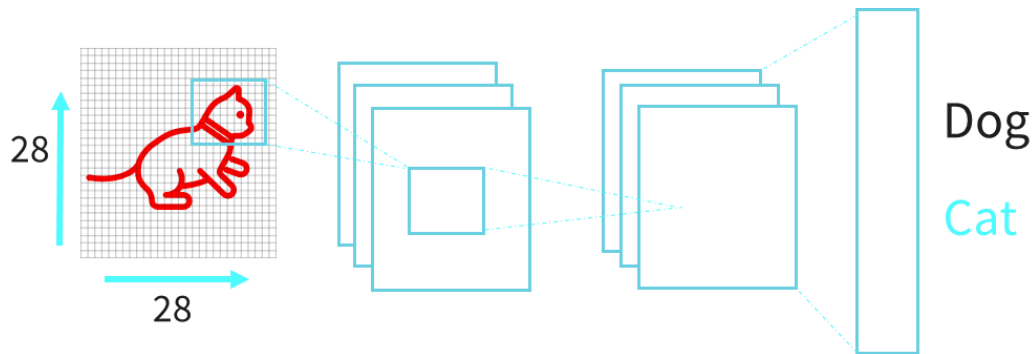
**Holberton**



# Why convolution & pooling in machine learning

# Why Convolution & Pooling for Machine Learning

- Building blocks of Convolutional Neural Networks
  - Widely used in image recognition, object detection
- Convolution helps in feature extraction
- Pooling helps to downsample data



# Background knowledge: image kernels

- Image kernel = small matrix of data
  - Commonly used in edge detection, blurring, or sharpening filters

```
208 205 247 240 244 253 247 245 136 151 255 255 255 255 255 255 254 257 231 255 254 254 255 255 254 255 252 255 255 254 255 247
244 161 137 244 254 255 254 255 118 103 259 228 155 153 236 130 14 52 88 173 255 254 254 255 255 255 254 255 254 253 244 184
102 154 75 200 240 255 255 255 110 96 84 81 31 44 85 53 44 45 43 54 140 213 253 255 255 255 255 245 157 186 176 223
90 108 96 143 223 255 255 252 117 75 41 35 31 24 25 38 45 44 44 48 81 118 148 254 252 254 255 248 231 248 255 254
67 89 107 156 255 255 255 255 104 25 34 35 25 21 25 34 32 33 32 34 53 85 100 142 231 242 247 240 255 255 255 255
55 51 45 134 215 251 255 252 51 12 28 33 24 24 48 75 82 78 71 88 58 53 67 93 136 228 258 156 253 246 240 255
79 58 96 75 224 255 255 118 11 27 74 99 91 106 140 162 173 173 173 172 156 137 92 48 78 157 217 206 254 222 233 255
38 43 47 52 147 255 229 96 41 81 129 145 160 169 169 172 175 175 175 175 177 177 172 110 31 62 259 236 255 244 240 255
40 40 33 38 90 245 171 32 85 110 139 145 151 162 171 174 176 179 182 184 187 183 173 162 71 45 167 255 254 255 254 255
37 44 44 31 89 250 158 36 70 129 143 142 153 162 171 175 177 178 182 191 194 188 180 170 120 51 137 255 254 250 254 255
34 45 51 64 155 237 161 53 116 136 140 143 154 164 176 178 174 177 183 186 185 185 183 178 140 88 141 254 252 225 240 255
34 38 52 74 71 188 156 83 131 134 144 155 160 161 173 179 178 179 189 193 190 185 167 162 156 93 148 250 254 214 247 255
32 38 52 74 71 188 156 83 131 134 144 155 160 161 173 179 178 179 189 193 190 185 167 162 156 93 148 250 254 214 247 255
32 38 52 74 71 188 156 83 131 134 144 155 160 161 173 179 178 179 189 193 190 185 167 162 156 93 148 250 254 214 247 255
38 32 72 120 212 228 115 85 121 104 102 104 94 103 134 156 170 162 125 108 121 143 155 180 191 104 134 230 253 253 255 251
81 82 116 107 179 247 134 80 101 90 111 119 103 81 94 147 191 118 128 88 123 153 147 181 200 92 100 222 257 167 227 215
144 116 167 231 210 232 170 67 115 88 76 82 83 85 88 139 162 150 135 80 53 99 141 165 201 97 79 192 245 235 246 240
127 145 140 156 254 213 157 95 133 122 117 133 126 106 110 139 191 197 167 129 127 148 147 171 188 110 121 226 253 180 215 212
87 112 100 79 85 82 85 75 142 148 151 153 138 125 120 149 191 190 193 175 174 193 196 208 127 163 239 219 149 156 195
83 83 109 134 129 106 59 78 132 142 155 159 130 111 124 164 155 250 186 192 191 195 250 252 250 143 217 250 249 242 238 234
88 78 78 113 97 74 43 106 127 140 152 155 125 97 112 150 185 154 174 183 196 186 252 258 259 186 247 254 255 254 254
72 44 83 59 46 52 49 74 127 137 140 149 132 103 78 90 134 141 168 165 159 257 254 253 216 193 236 244 251 242 236 243
55 20 89 73 59 80 46 74 117 127 144 161 148 124 106 120 196 167 193 162 189 206 201 255 214 194 174 165 197 188 183 193
85 49 77 89 50 68 43 81 109 127 147 113 100 121 145 145 169 161 176 161 201 201 255 252 174 186 169 176 183 158 194
82 76 92 79 54 58 37 47 90 121 132 116 89 78 111 146 163 140 122 124 180 197 197 196 176 149 146 152 155 157 159 168
104 107 122 125 79 27 33 86 111 122 120 114 114 147 175 190 186 163 101 170 200 187 185 156 146 145 130 137 141 140 145
117 124 127 133 135 105 21 28 37 86 115 121 126 126 141 142 188 252 212 153 164 186 180 188 154 146 144 140 151 151 147 144
119 118 118 125 128 111 21 28 58 100 118 131 140 151 159 186 201 255 162 180 188 149 186 119 144 147 143 140 141 144 148
117 119 125 130 139 106 18 29 44 58 70 102 133 147 168 197 212 215 210 136 177 152 133 105 57 59 126 151 145 143 142 141
115 123 126 134 145 102 27 54 52 38 45 89 105 135 175 189 193 216 208 188 139 111 164 203 74 5 121 151 142 142 143 146
101 108 123 121 132 125 44 40 31 35 57 44 58 101 147 144 136 163 145 94 90 145 196 167 84 45 185 180 142 144 142 145
98 97 97 98 104 76 34 33 30 48 41 49 51 58 74 53 95 88 83 89 190 188 250 156 162 106 140 140 125 133 131 131
102 102 97 88 73 35 30 23 42 50 66 41 90 80 59 51 57 82 157 157 255 169 82 98 151 105 101 154 135 130 129
```



A solid red vertical bar is positioned to the left of the text.

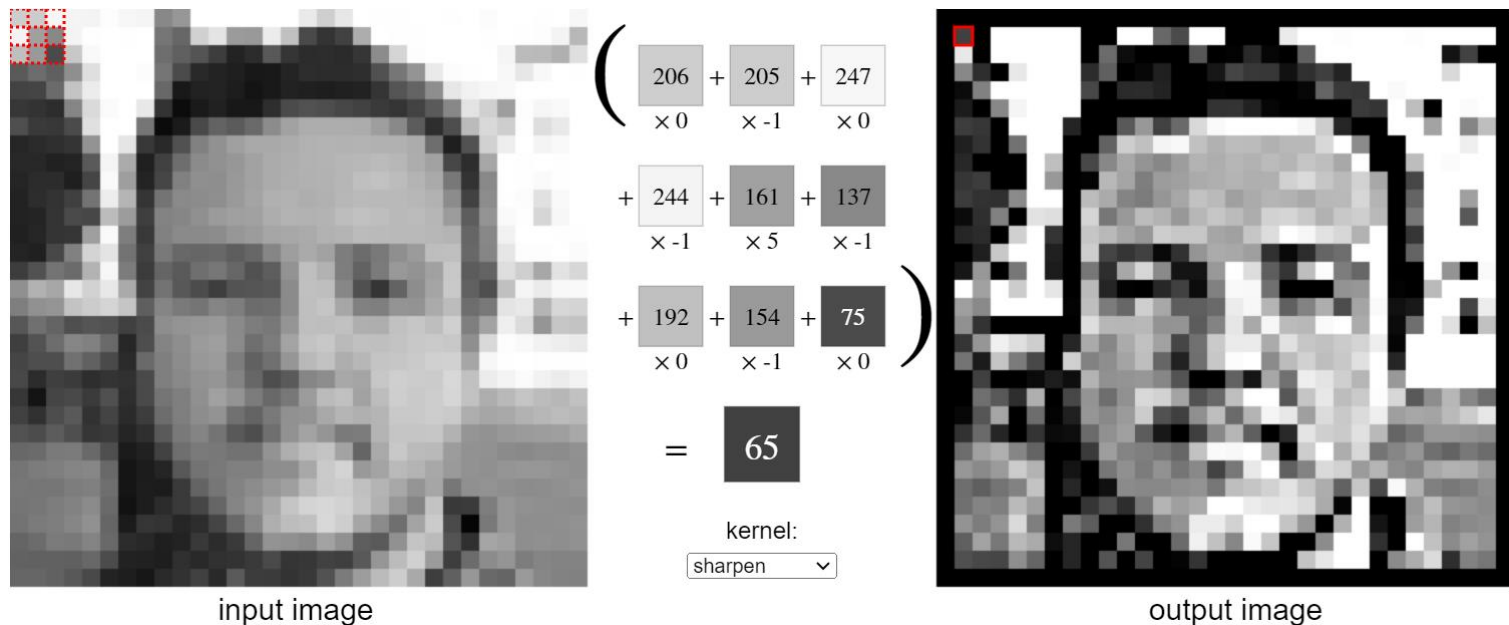
**What is convolution?**

# What is convolution

- Convolution is a mathematical operation where a filter slides over input data
  - It computes the **element-wise** multiplication between the filter and input at each position
- Convolution is used in feature extraction
  - Input data is usually a matrix representing an image
- Filter is a smaller matrix that slides over the input to extract features
  - The element-wise multiplication results in a new matrix called **feature map**

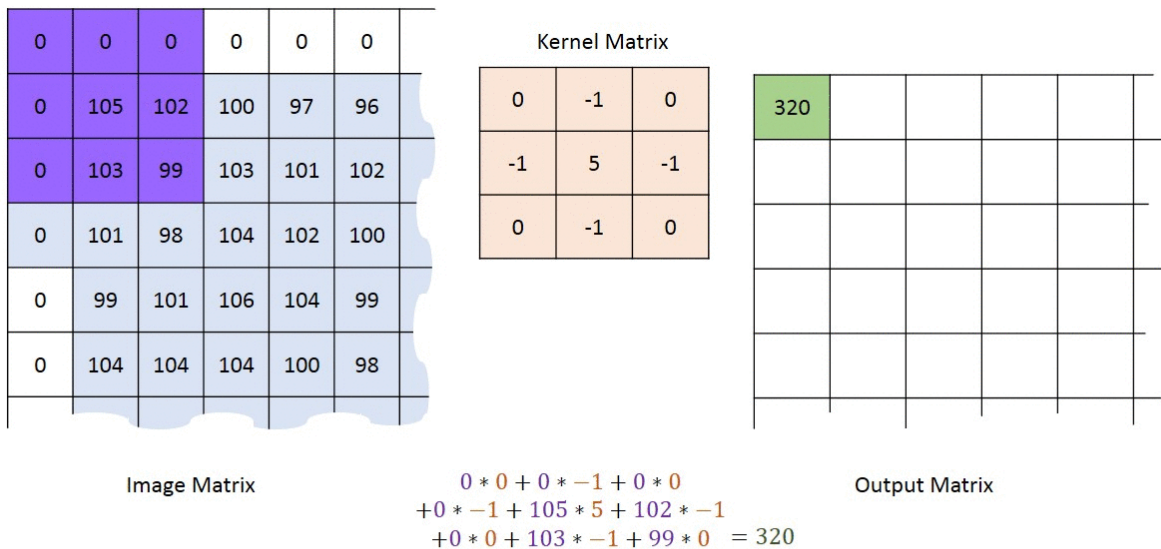
# What is convolution

- Convolution is a mathematical operation where a filter slides over input data



# What is convolution

- Convolution is a mathematical operation where a filter slides over input data



Convolution with horizontal and  
vertical strides = 1





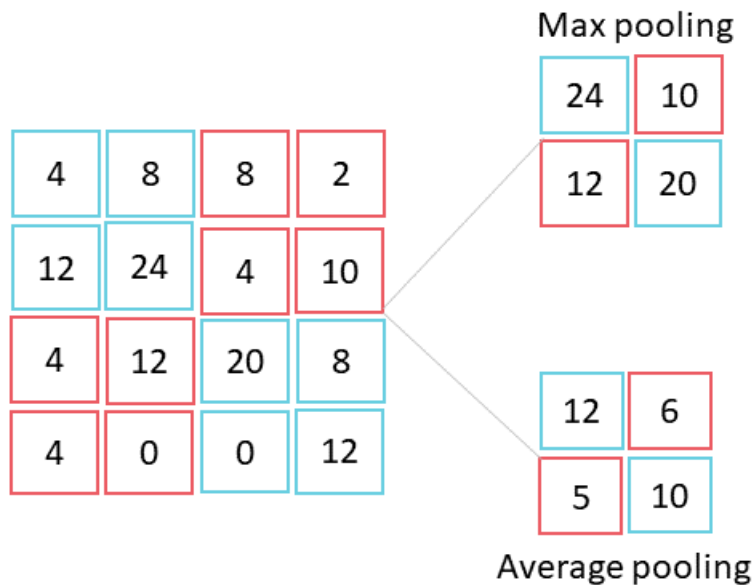
**What is pooling?**

# | What is pooling

- Pooling refers to a down-sampling process
  - Reduces the spatial dimensions of the feature maps produced by convolution
- Pooling is typically applied after a convolution layer
- Pooling helps to prevent “overfitting,,
  - Reduces the dimensionality of the feature map & parameters of the model
- Pooling makes the model more robust to small variations
  - ...but can result in loss of information

# | Types of pooling

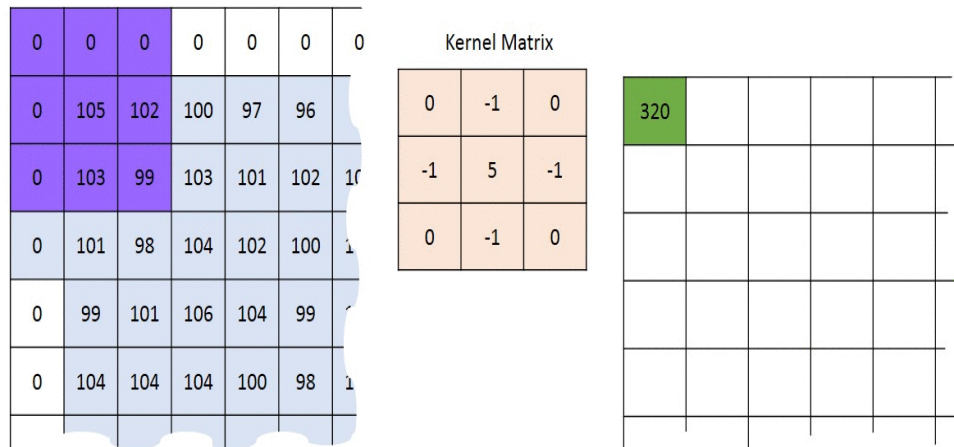
- Pooling divides the input image / feature map into small non-overlapping regions
  - ...then it replaces each region with a single value
- **Max pooling:**
  - takes maximum value in region
- **Average pooling:**
  - takes average value in region



# Fundamental components of convolution & pooling

# Fundamental elements: stride

- **Stride** = number of pixels by which the kernel slides over the input
- A stride of 1 means the kernel moves one pixel at a time
- Increasing the stride reduces the spatial dimensions
- Smaller stride provides fine-grained output feature map

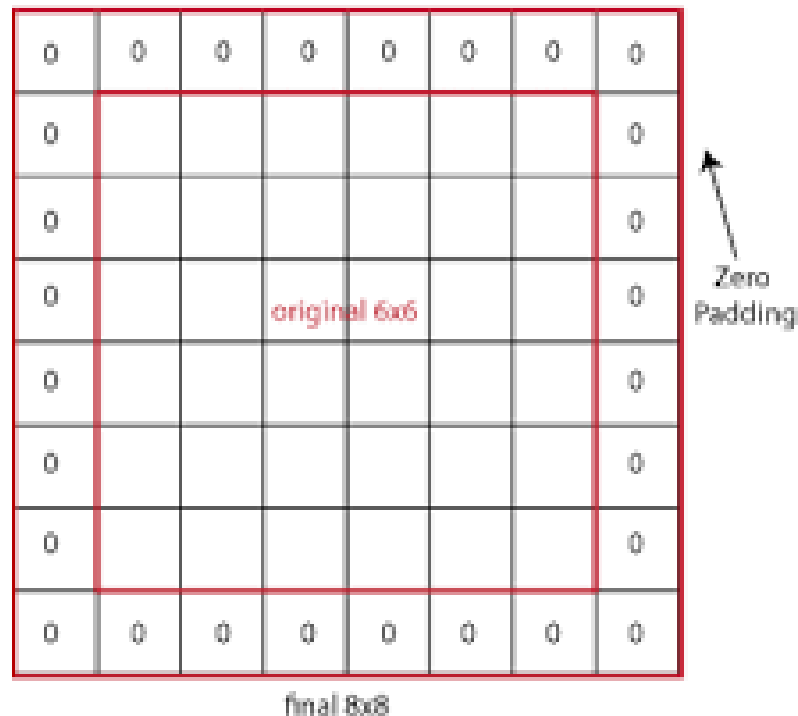


$$\begin{aligned} &0 * 0 + 0 * -1 + 0 * 0 \\ &+ 0 * -1 + 105 * 5 + 102 * -1 \\ &+ 0 * 0 + 103 * -1 + 99 * 0 = 320 \end{aligned}$$

Convolution with horizontal and  
vertical strides = 2

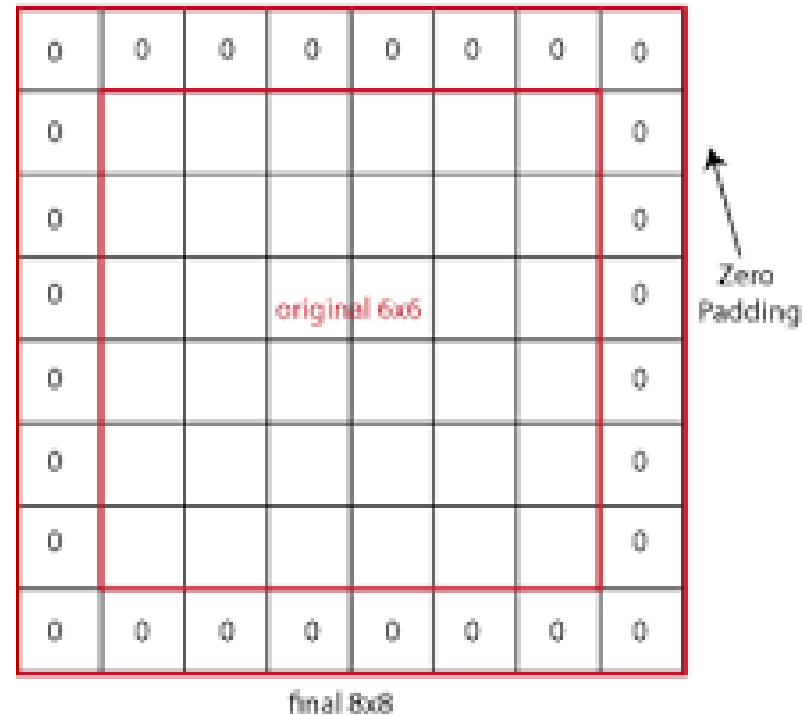
# Fundamental elements: padding

- **Padding** = ensure feature map has same dimensions as input
- Padding adds extra rows and cols of zeroes around the edge of input image or feature map
- When no padding is used, the kernel may not slide all the way to the edge



## Fundamental elements: types of padding

- **Zero padding:** zeroes are added around the edge of the input image
- **Same padding:** padding is added to input to match spatial dimensions of output
- Many other types of padding exist



## Fundamental elements: feature map

- **Feature map** = 3D array of values representing the output of a convolutional layer
- Each value corresponds to a particular region in input image
- Number of feature maps corresponds to number of kernels

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

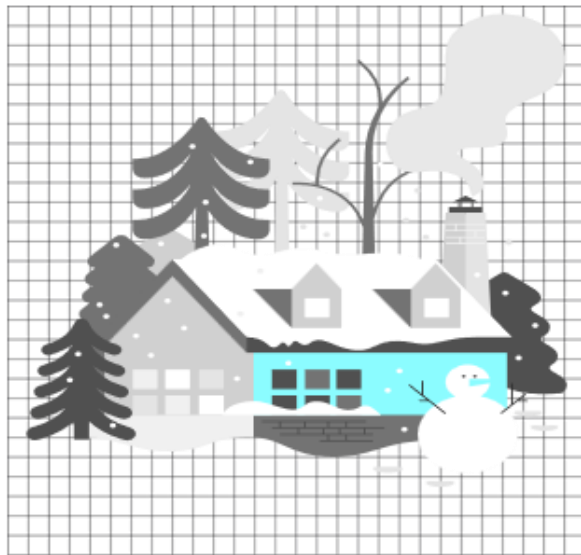




# Convolution over an image

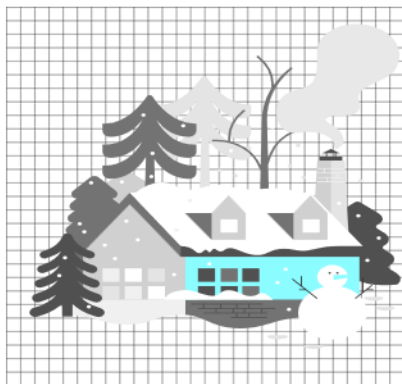
# How to detect objects in images?

- Images are represented as layers of pixel values
- Rely on convolution operations to detect features
- Rely on pooling operations to reduce complexity



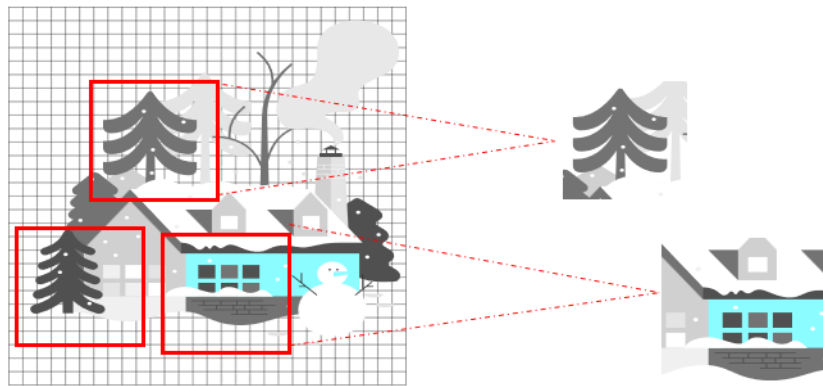
# How to detect objects in images?

- Images are represented as layers of pixel values
- Each pixel denotes the value of a color channel (0..255)

[illegible]

# How to detect objects in images?

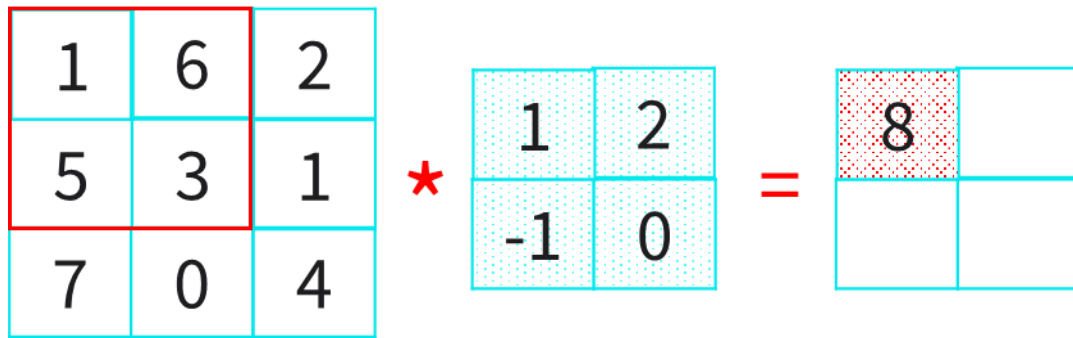
- The convolution layer



- Mathematical operation on input
  - Relies on input, filter & feature map

# How to detect objects in images?

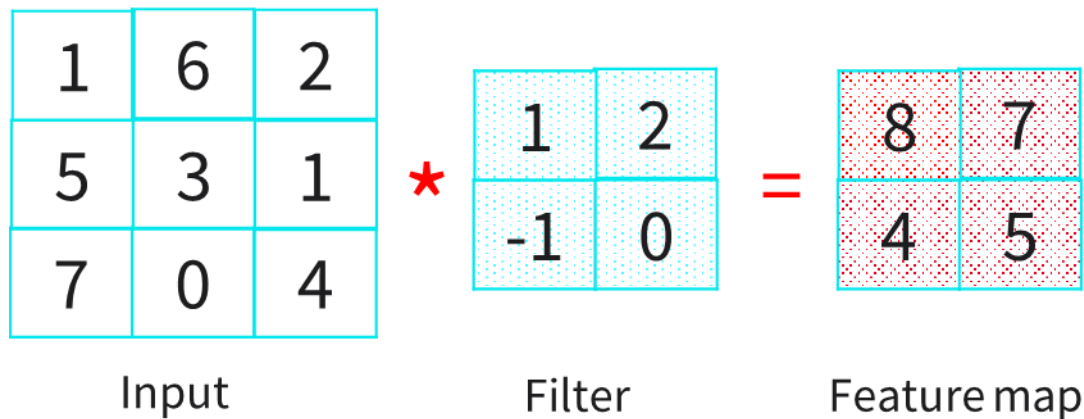
- Convolution & filter
- Example:  $1 \cdot 1 + 2 \cdot 6 + (-1) \cdot 5 + 0 \cdot 3 = 8$



- Filter moves across an image and checks if a feature is present

# How to detect objects in images?

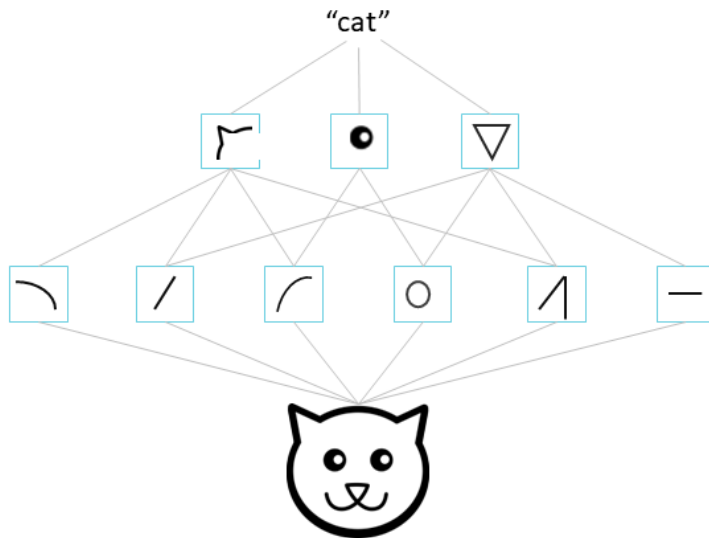
- Convolution & filter



- Filter moves across an image and checks if a feature is present

# How to detect objects in images?

- Visual spatial hierarchy example



- Patterns not affected by rotation & translation
- Builds up layer of abstraction



**Any questions?**

