ENGR 3321: Introduction to Deep Learning for Robotics

Neural Network NN1:

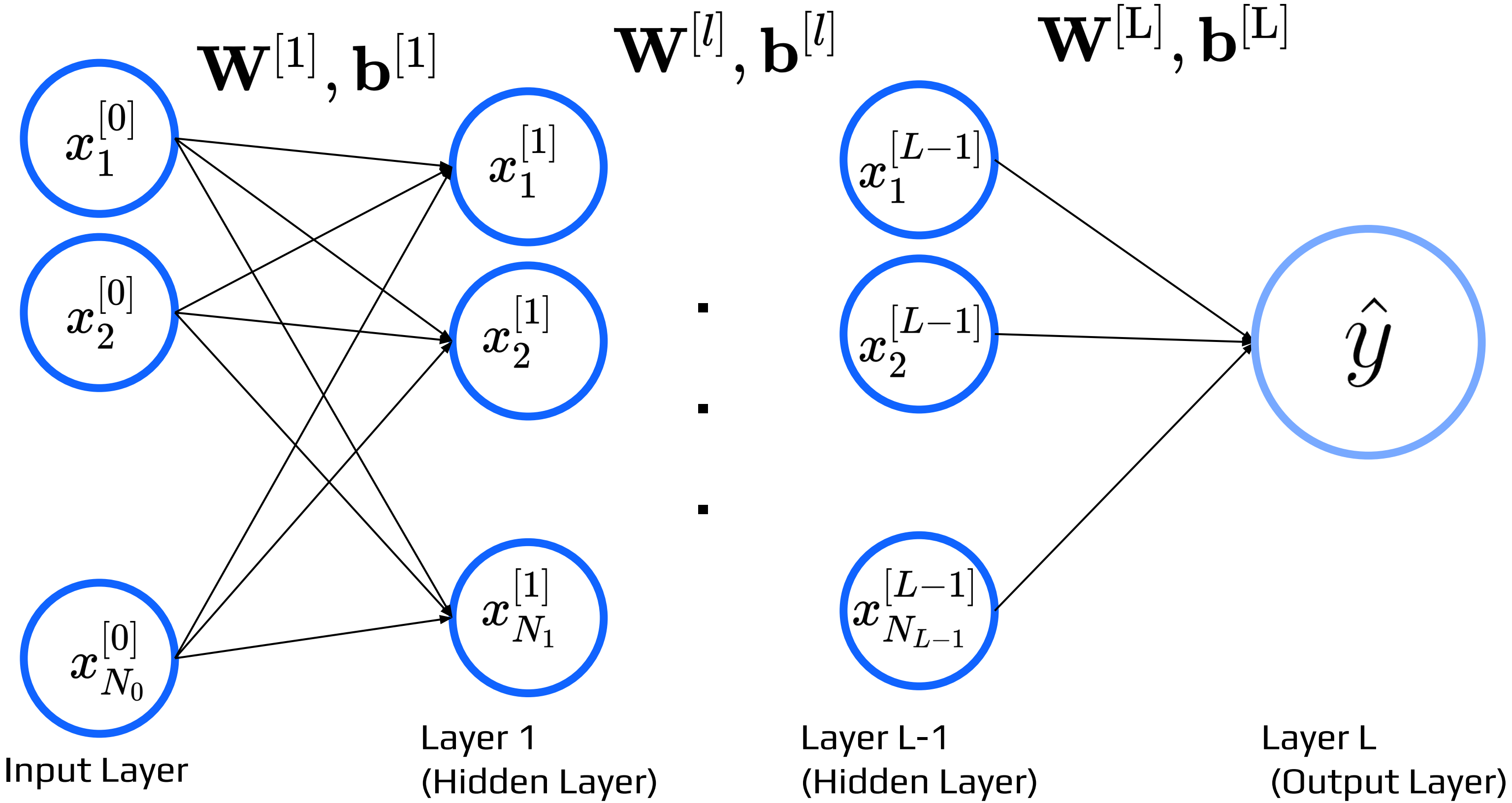
Multi-Input, Multi-Hidden Layer, One-Output Model

10/06/2025

Outline

- Multi input, multi hidden layer, single output Model
- Image Processing

Graphical Representation



Review: Model Training

1. Prepare datasets: train, validation
2. (Randomly) Initialize model parameters: w, b.
3. Evaluate the model with a metric (e.g. BCE, MSE).
4. Calculate gradients of loss.
5. Update parameters a small step on the directions descending the gradient of loss.
6. Repeat 3 to 5 until converge.

Prepare Datasets: Training

A dataset with M_{tr} samples:

- Each sample has N features: $x_1, x_2, \ldots, x_N$
- Each sample is labeled: y ($y \in \{0, 1\}$ for binary classification)

$$\mathcal{D} = \{({}^{(1)}x_1^{[0]}, {}^{(1)}x_2^{[0]}, \ldots, {}^{(1)}x_N^{[0]}, {}^{(1)}y), ({}^{(2)}x_1^{[0]}, {}^{(2)}x_2^{[0]}, \ldots, {}^{(2)}x_N^{[0]}, {}^{(2)}y), \ldots, ({}^{(M_{tr})}x_1^{[0]}, {}^{(M_{tr})}x_2^{[0]}, \ldots, {}^{(M_{tr})}x_N^{[0]}, {}^{(M_{tr})}y)\}$$

$$= \{({}^{(1)}\mathbf{x}^{[0]}, {}^{(1)}y), ({}^{(2)}\mathbf{x}^{[0]}, {}^{(2)}y), \ldots, ({}^{(M_{tr})}\mathbf{x}^{[0]}, {}^{(M_{tr})}y)\}$$

Prepare Datasets: Validation

A dataset with M_v ($M_v < M_{tr}$) samples:

- Each sample has N features: $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N$
- Each sample is labeled: y
- Validation dataset can be used to evaluate model.
- Validation dataset does not participate into model updating

$$\mathcal{D} = \{({}^{(1)}\tilde{x}_1, {}^{(1)}\tilde{x}_2, \ldots, {}^{(1)}\tilde{x}_N, {}^{(1)}y), ({}^{(2)}\tilde{x}_1, {}^{(2)}\tilde{x}_2, \ldots, {}^{(2)}\tilde{x}_N, {}^{(2)}y), \ldots, ({}^{(M_v)}\tilde{x}_1, {}^{(M_v)}\tilde{x}_2, \ldots, {}^{(M_v)}\tilde{x}_N, {}^{(M_v)}y)\}$$

$$= \{({}^{(1)}\tilde{\mathbf{x}}, {}^{(1)}y), ({}^{(2)}\tilde{\mathbf{x}}, {}^{(2)}y), \ldots, ({}^{(M_v)}\tilde{\mathbf{x}}, {}^{(M_v)}y)\}$$

NN1 Model (Matrix Representation)

$$\hat{\mathbf{y}} = \sigma(\mathbf{X}^{[L-1]} \cdot \mathbf{W}^{[L]T} + \mathbf{b}^{[L]})$$

$(M,1) \quad (M,N_{L-1}) \quad (N_{L-1},1) \quad (1,1)$

$$\boldsymbol{X}^{[l]}=\sigma(\boldsymbol{X}^{[l-1]}\cdot\boldsymbol{W}^{[l]T}+\boldsymbol{b}^{[l]})\,,\,l=1,2,\ldots,L-1$$

$(M,N_l) \quad (M,N_{l-1}) \quad (N_{l-1},N_l) \quad (1,N_l)$

Model Evaluation Metrics

Binary Cross Entropy (BCE)

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^{M} -{}^{(i)}y \ln {}^{(i)}\hat{y} - (1 - {}^{(i)}y) \ln(1 - {}^{(i)}\hat{y}) = \overline{-\mathbf{y} \ln \hat{\mathbf{y}} - (1 - \mathbf{y}) \ln(1 - \hat{\mathbf{y}})}$$

Mean Squared Error (MSE)

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)^2 = \overline{(\hat{\mathbf{y}} - \mathbf{y})^2}$$

Gradients of Loss

$$\nabla\mathcal{L} = \left[\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}^{[1]}}\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{b}^{[1]}} \cdots \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}^{[l]}}\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{b}^{[l]}} \cdots \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}^{[L]}}\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{b}^{[L]}}\right]$$

Back-Propagation (Last Layer)

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{[L]}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{Z}^{[L]}} \frac{\partial \mathbf{Z}^{[L]}}{\partial \mathbf{W}^{[L]}} = \frac{1}{M} (\hat{\mathbf{y}} - \mathbf{y})^T \cdot \mathbf{X}^{[L-1]}$$

$$\frac{\partial \mathcal{L}}{\partial b^{[L]}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{Z}^{[L]}} \frac{\partial \mathbf{Z}^{[L]}}{\partial b^{[L]}} = \overline{\hat{\mathbf{y}} - \mathbf{y}}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[L-1]}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{Z}^{[L]}} \frac{\partial \mathbf{Z}^{[L]}}{\partial \mathbf{X}^{[L-1]}} = (\hat{\mathbf{y}} - \mathbf{y}) \cdot \mathbf{W}^{[L]}$$

Back-Propagation (Hidden Layers)

$l = 1, \ldots, L-1$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{[l]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l]}} \frac{\partial \mathbf{X}^{[l]}}{\partial \mathbf{Z}^{[l]}} \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{W}^{[l]}} = \frac{1}{M} [\frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l]}} * \mathbf{X}^{[l]} * (1 - \mathbf{X}^{[l]})]^T \cdot \mathbf{X}^{[l-1]}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{[l]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l]}} \frac{\partial \mathbf{X}^{[l]}}{\partial \mathbf{Z}^{[l]}} \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{b}^{[l]}} = \overline{\frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l]}} * \mathbf{X}^{[l]} * (1 - \mathbf{X}^{[l]})}, \text{ axis} = 0, \text{ keepdim}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l-1]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l]}} \frac{\partial \mathbf{X}^{[l]}}{\partial \mathbf{Z}^{[l]}} \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{X}^{[l-1]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{X}^{[l]}} * \mathbf{X}^{[l]} * (1 - \mathbf{X}^{[l]}) \cdot \mathbf{W}^{[l]}$$

BP stops when $l = 1$

Gradient Descent Optimization

Given dataset: $\left\{ \left({}^{(1)}\mathbf{x}, {}^{(1)}\mathbf{y} \right), \left({}^{(2)}\mathbf{x}, {}^{(2)}\mathbf{y} \right), \ldots, \left({}^{(M)}\mathbf{x}, {}^{(M)}\mathbf{y} \right) \right\}$

Initialize $\mathbf{W}^{[l]},\ \mathbf{b}^{[l]}$

Repeat until converge {

compute $\mathcal{L}\left(\hat{\mathbf{Y}}, \mathbf{Y}\right)$

compute $\nabla\mathcal{L}$

$\mathbf{W}^{[l]} := \mathbf{W}^{[l]} - \alpha \cdot d\mathbf{W}^{[l]}$

$\mathbf{b}^{[l]} := \mathbf{b}^{[l]} - \alpha \cdot d\mathbf{b}^{[l]}$

}

where α is learning rate

Color Image

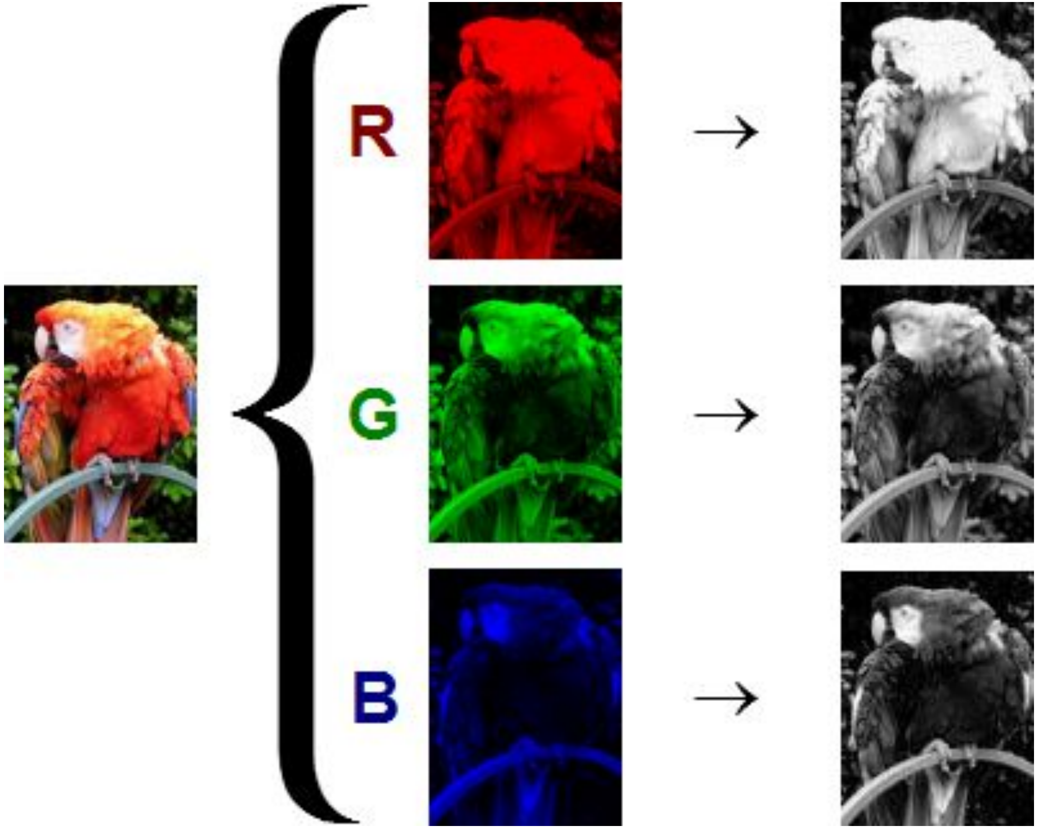


Image Representation

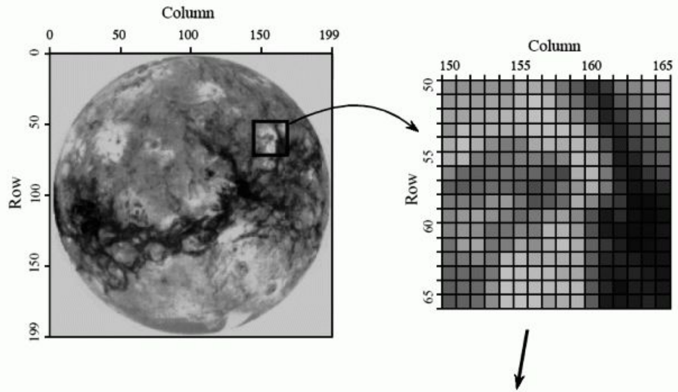


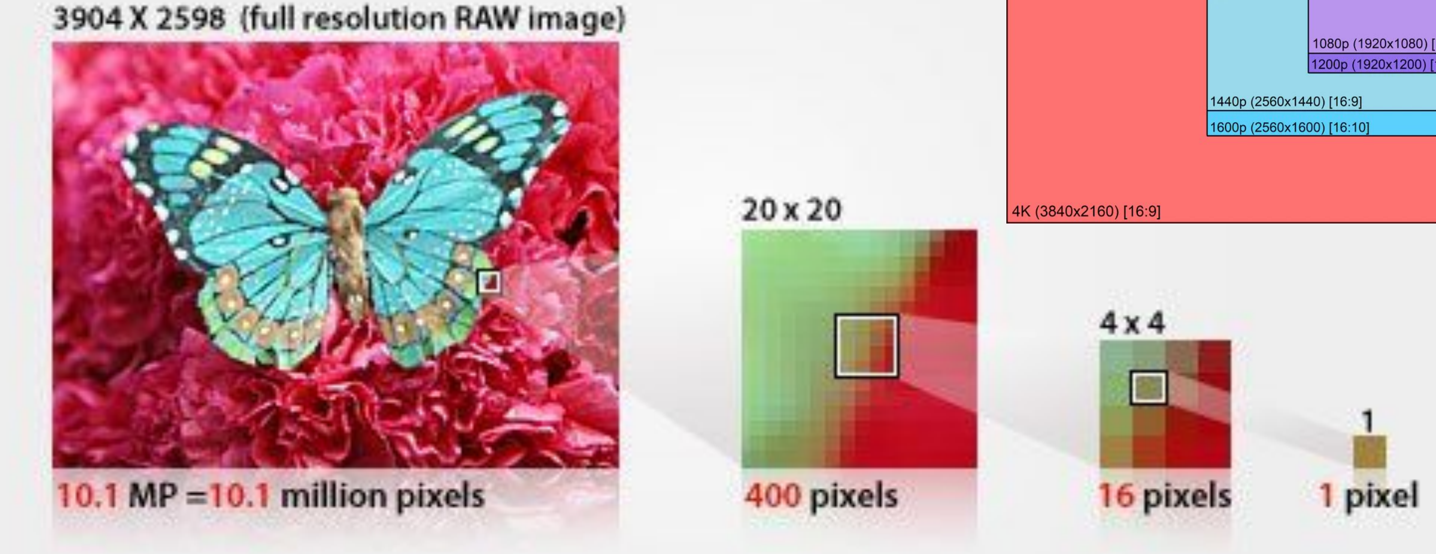
FIGURE 23-1
Digital image structure. This example image is the planet Venus, as viewed in reflected microwaves. Digital images are represented by a two-dimensional array of numbers, each called a *pixel*. In this image, the array is 200 rows by 200 columns, with each pixel a number between 0 to 255. When this image was acquired, the value of each pixel corresponded to the level of reflected microwave energy. A *grayscale* image is formed by assigning each of the 0 to 255 values to varying shades of gray.

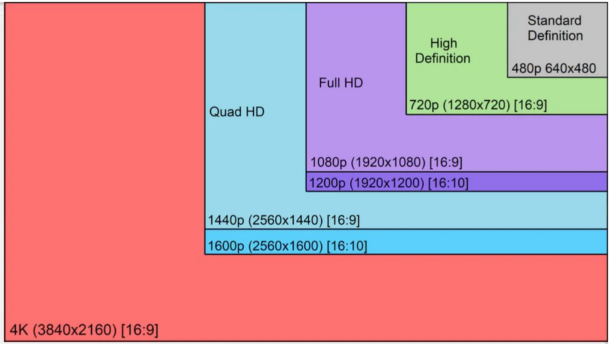
Column

Row	150					155					160					165
50	183	183	181	184	177	200	200	189	159	135	94	105	160	174	191	196
	186	195	190	195	191	205	216	206	174	153	112	80	134	157	174	196
	194	196	198	201	206	209	215	216	199	175	140	77	106	142	170	186
	184	212	200	204	201	202	214	214	214	205	173	102	84	120	134	159
	202	215	203	179	165	165	199	207	202	208	197	129	73	112	131	146
55	203	208	166	159	160	168	166	157	174	211	204	158	69	79	127	143
	174	149	143	151	156	148	146	123	118	203	208	162	81	58	101	125
	143	137	147	153	150	140	121	133	157	184	203	164	94	56	66	80
	164	165	159	179	188	159	126	134	150	199	174	119	100	41	41	58
	173	187	193	181	167	151	162	182	192	175	129	60	88	47	37	50
60	172	184	179	153	158	172	163	207	205	188	127	63	56	43	42	55
	156	191	196	159	167	195	178	203	214	201	143	101	69	38	44	52
	154	163	175	165	207	211	197	201	201	199	138	79	76	67	51	53
	144	150	143	162	215	212	211	209	197	198	133	71	69	77	63	53
	140	151	150	185	215	214	210	210	211	209	135	80	45	69	66	60
65	135	143	151	179	213	216	214	191	201	205	138	61	59	61	77	63

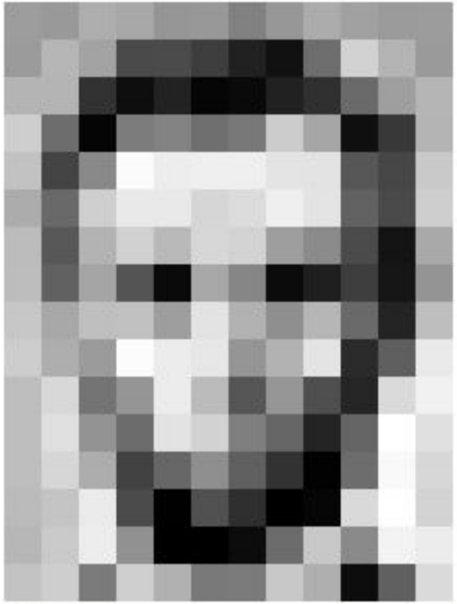
Chapter 23, The scientist and engineer's guide to digital signal processing

Image Resolution





Pixel Intensity



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Pixel Localization

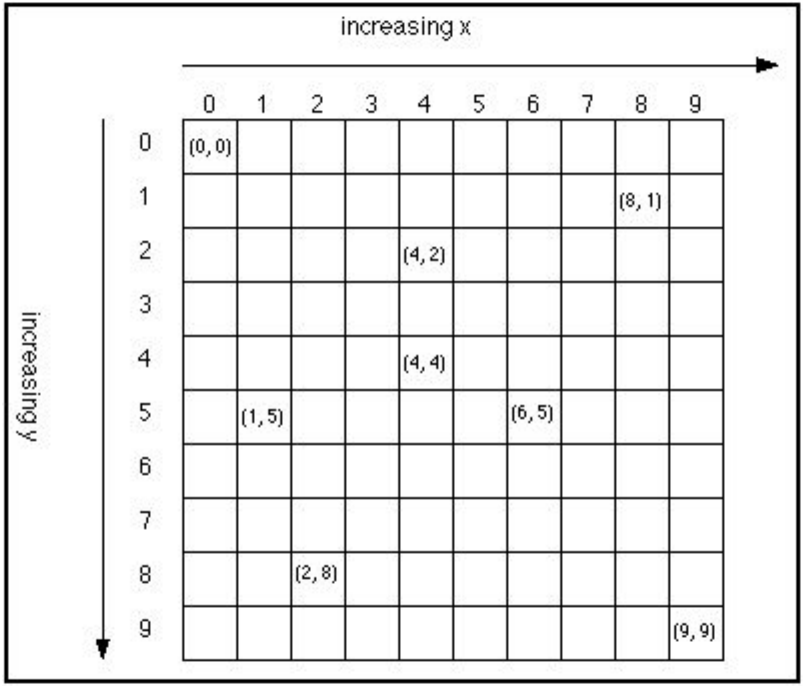


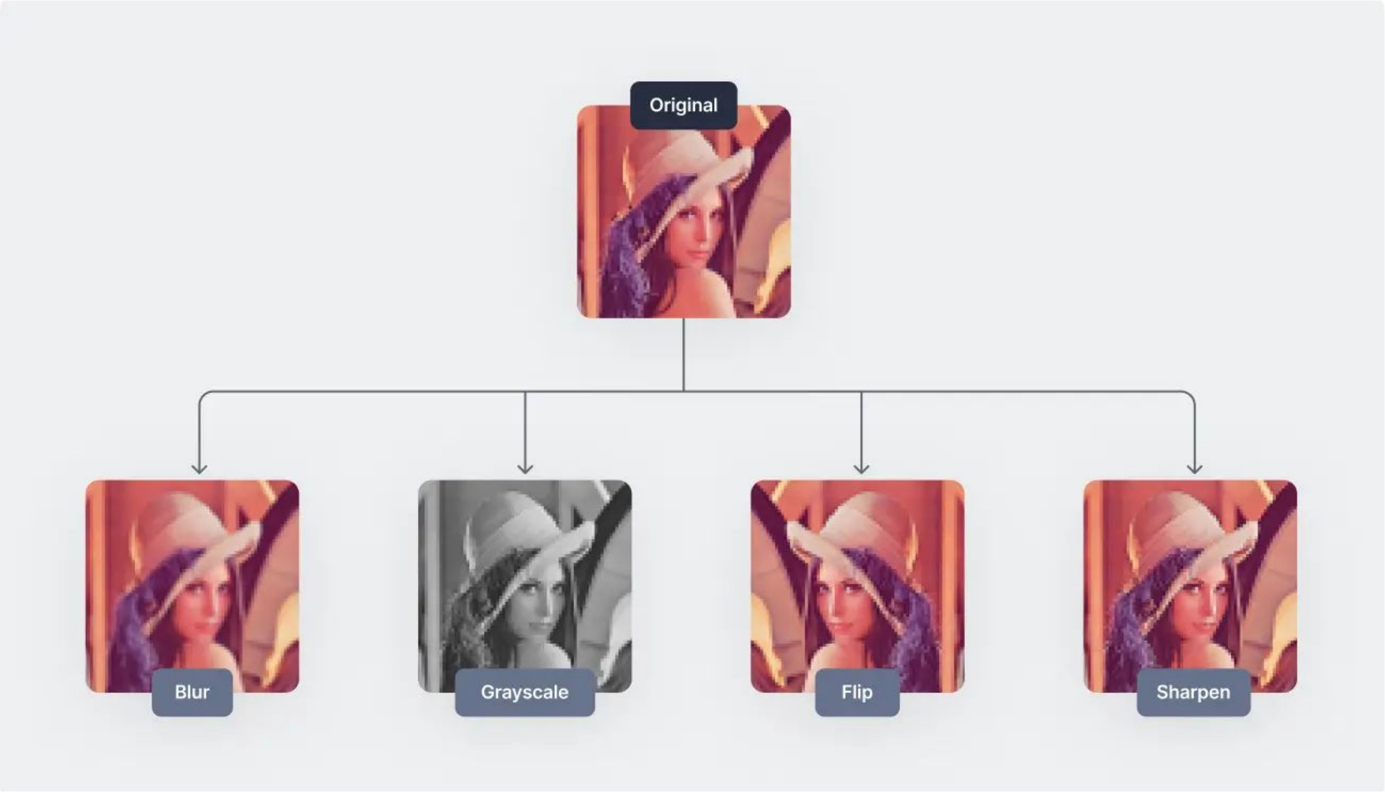
Image File Formats

web, screen printing

image format	colour model	transparency	destination	remarks
JPG	RGB	—		generational degradation
TIFF	RGB / CMYK	✓		layered images, image stacks
GIF	RGB	✓		limited colour, animated images
PNG	RGB	✓		lossless compression

file format	colour model	transparency	destination	remarks
SVG	RGB	✓		interactive, scriptable
EPS	RGB / CMYK	✓		PostScript document
PDF	RGB / CMYK	✓		includes PostScript, platform independent

Low-Level Image Processing

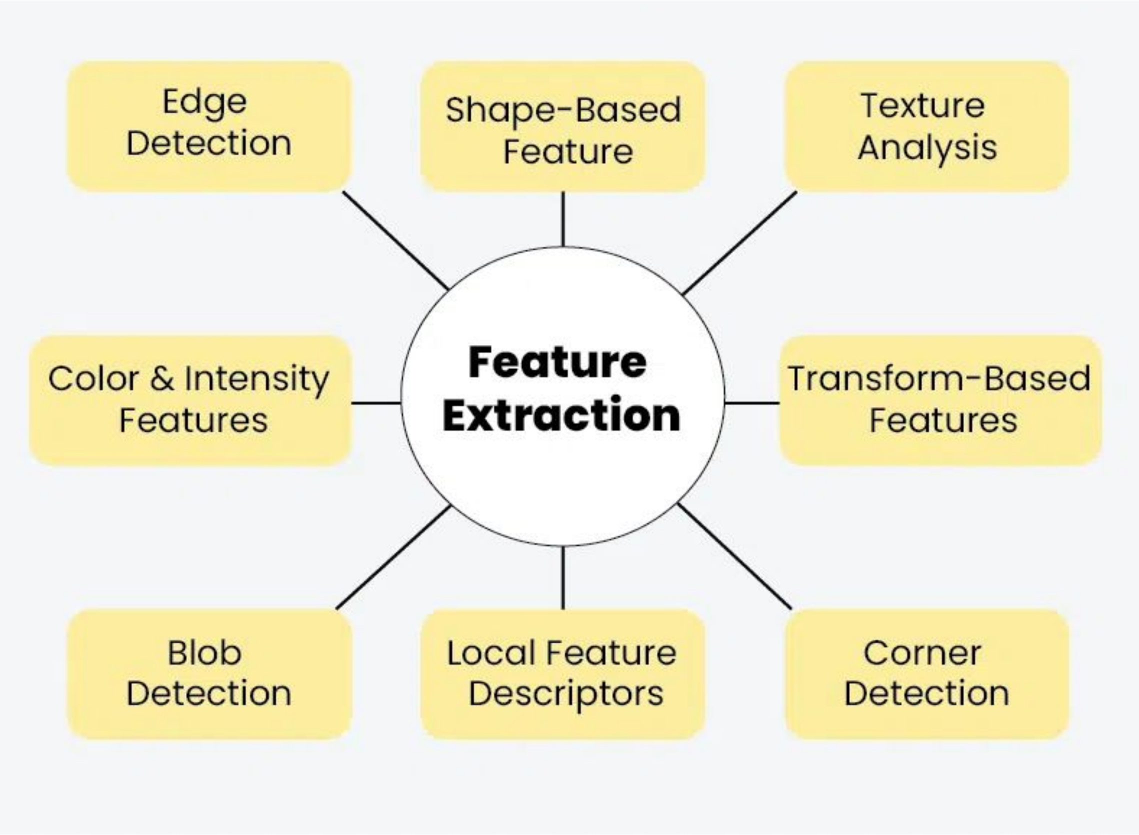


High-Level Image Processing

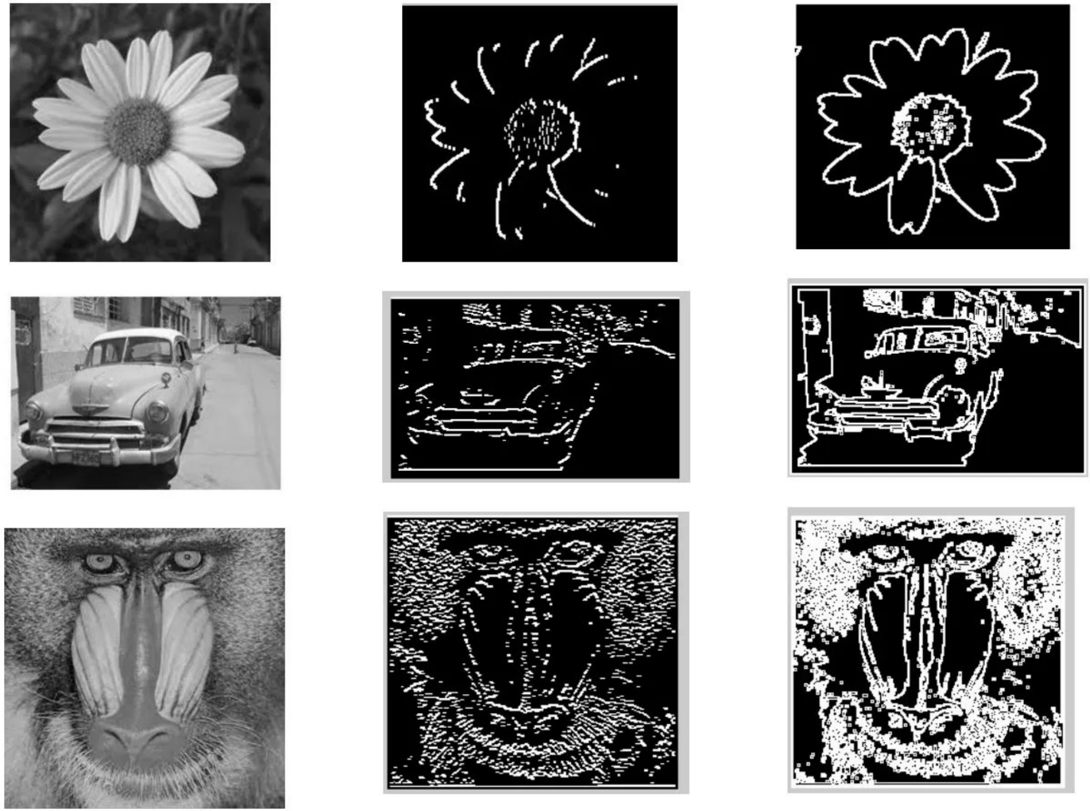
- Image restoration
- Object detection and recognition
- Image enhancement
- Image segmentation
- Feature extraction
- Morphological processing
- Analogue image processing
- Image compression
- Pattern recognition

…

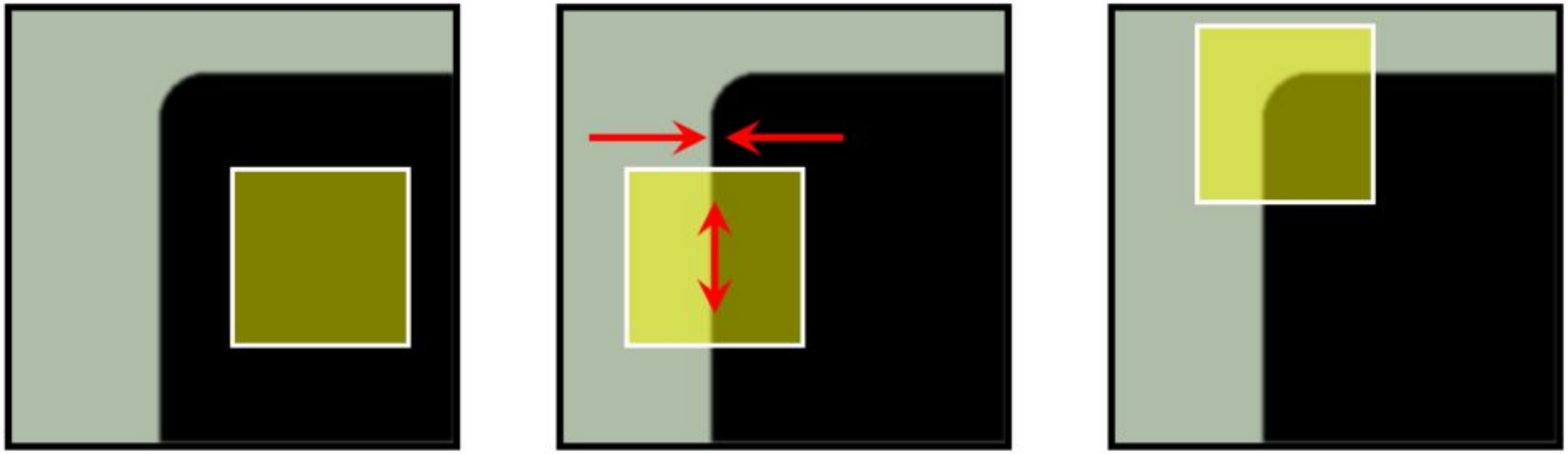
Image Feature Extraction



Edge Detection



Corner Detection



"flat" region: no change in all directions

"edge" : no change along the edge direction

"corner" : significant change in all directions with small shift

Feature Transform

Color Histogram

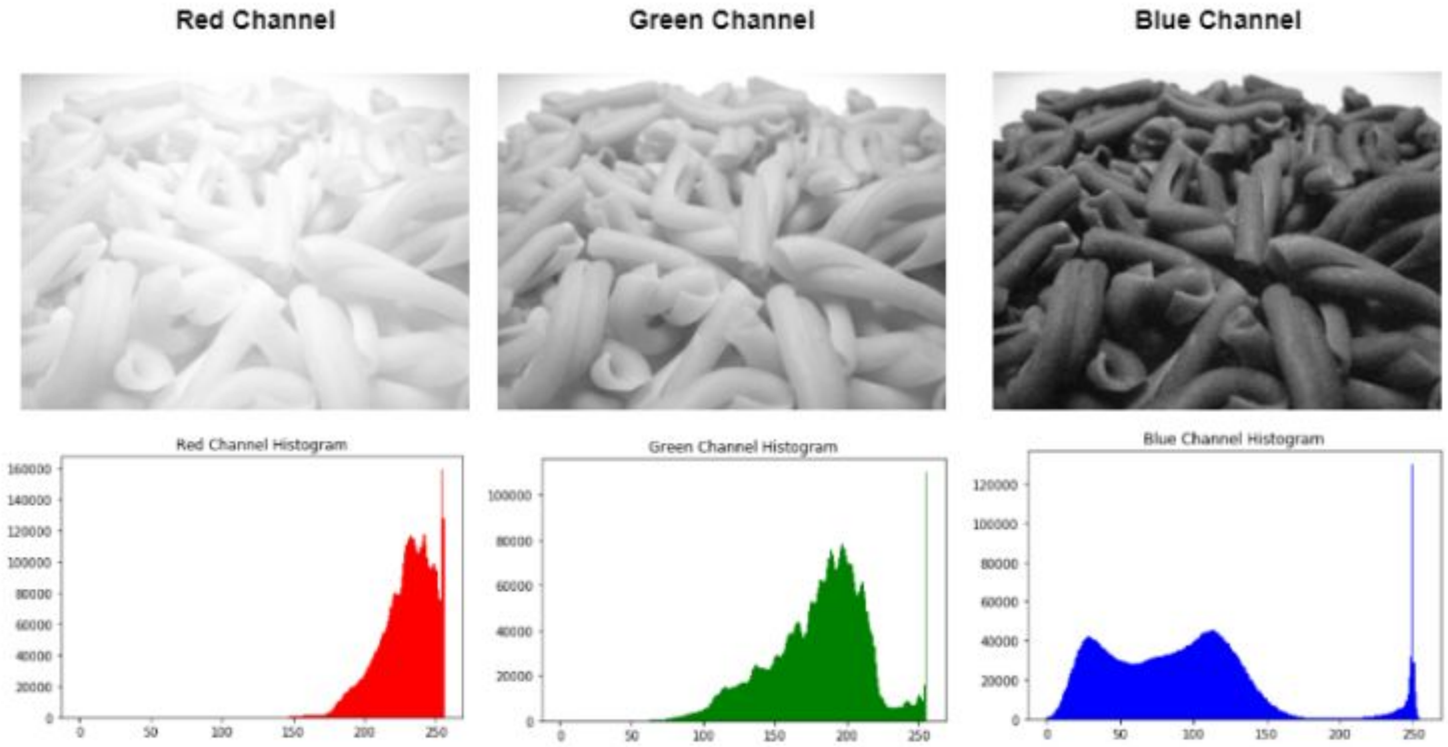
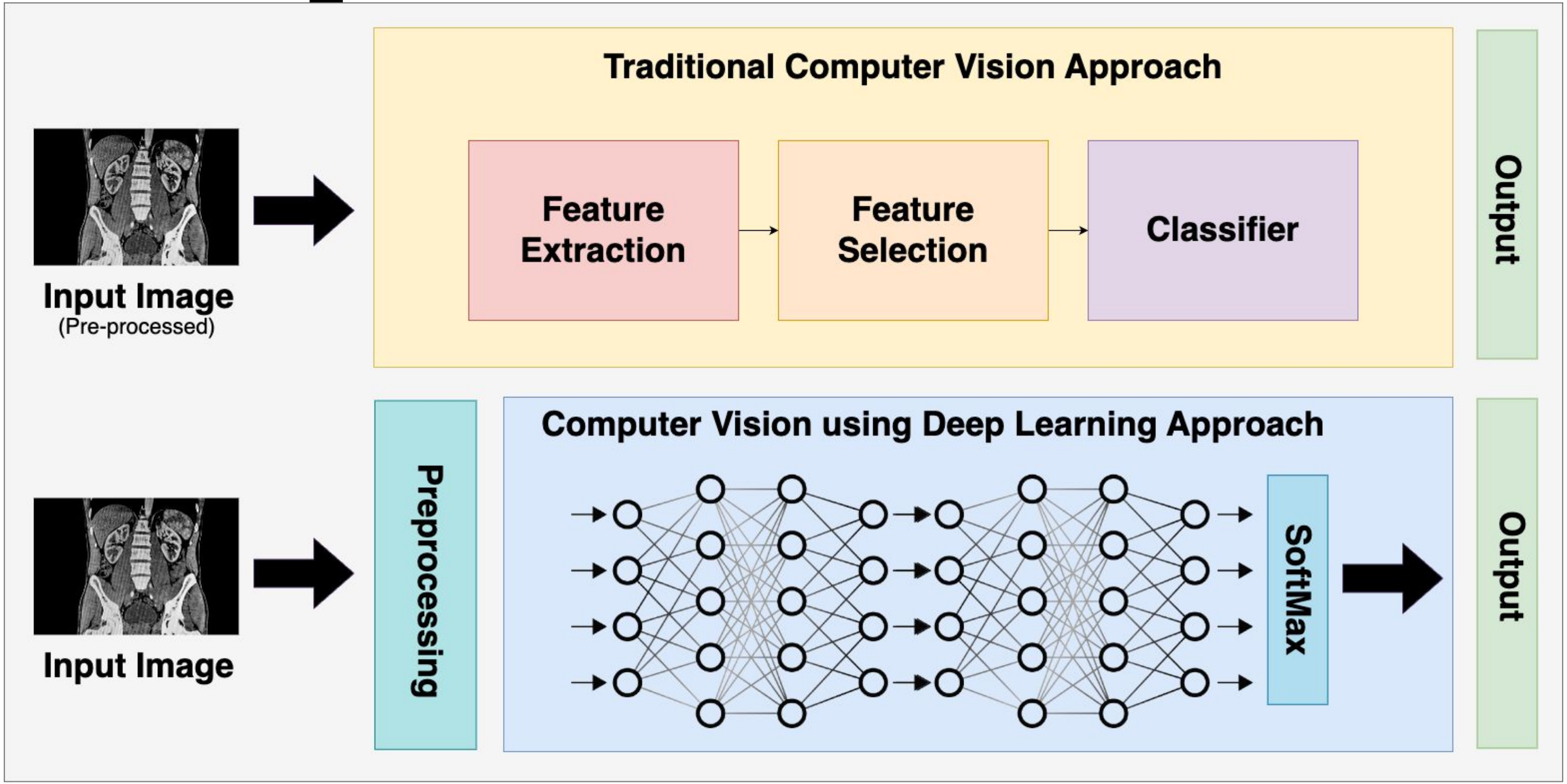


Image Processing w/ Deep Learning



Deep Learning Advantages

- Automatic Feature Learning
  - Traditional Techniques: Require handcrafted features that are manually designed by experts.
  - **Deep Learning**: Automatically learns the optimal features directly from raw data (e.g., pixel values).
- Better Performance on Complex Tasks
  - Traditional Techniques: Work well on relatively simple, controlled datasets.
  - **Deep Learning**: Excels on complex datasets with many variations (e.g., those with high variability in lighting, object orientation, or backgrounds).
- End-to-End Learning
  - Traditional Techniques: Involve separate stages — first, feature extraction , and then classification.
  - **Deep Learning**: Provides an end-to-end learning process, meaning that the entire model (from raw input to output) is optimized in one step.
- Scalability and Adaptability
  - Traditional Techniques: Often need significant adjustments when applied to different tasks.
  - **Deep Learning**: Deep learning models are highly scalable and adaptable across different image types and tasks.