

1.Cvičení

Základy prostředí MATLAB

Ing. Ilona Janáková, PhD.

SE2.144

janakova@vutbr.cz

Ing. Lukáš Kratochvíla

SE2.145

kratochvila@vutbr.cz

Ing. Dominik Řičánek

SE2.145

203334@vut.cz

Organizace

- ▶ Účast na cvičeních povinná
- ▶ Možnost přijít na cvičení v jiný termín (vyjíměně)
- ▶ 15 bodů za práci na cvičení (+ 15 bodů od doc. Jirsíka):
 - ▶ 2. cvičení - nula bodů
 - ▶ 3.-5. cvičení - za úspěšné vypracování zadání až 4 body
 - ▶ 11. cvičení - za úspěšné vypracování zadání až 3 body
- ▶ Zápočet: minimálně 15 bodů (cvičení + projekty)

Osnova cvičení

- ▶ Základy Matlabu (0 bodů, 2. týden)
- ▶ Perceptron (4 body, 3. týden)
- ▶ Jednovrstvá síť (4 body, 4. týden)
- ▶ Vícevrstvá síť (4 body, 5. týden)
- ▶ Základy počítačového vidění, 2D konvoluce (3 body, 11. týden)

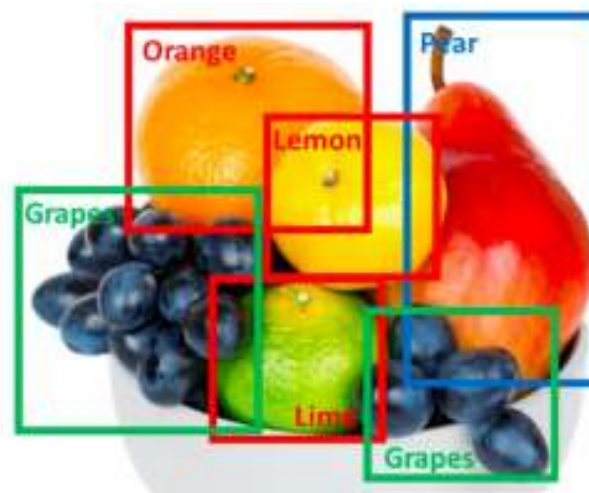
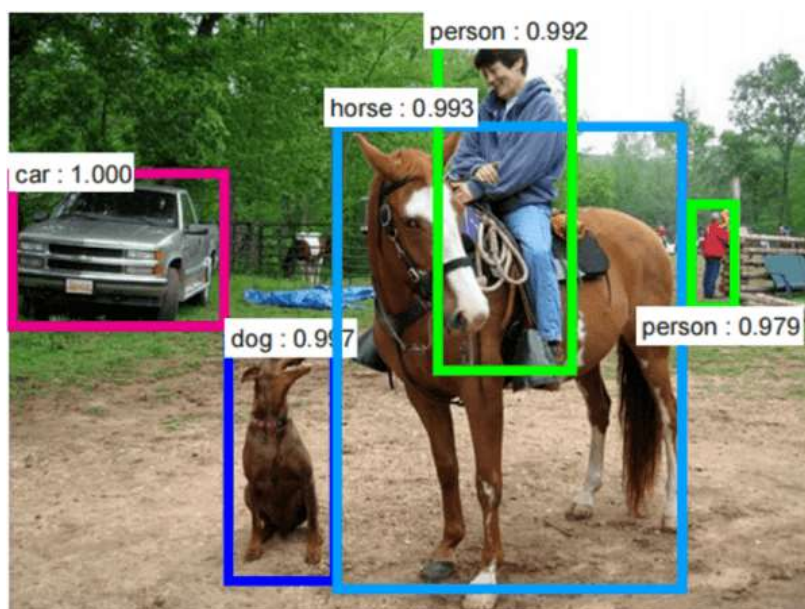
Aplikace umělých neuronových sítí

- ▶ **Zpracování obrazu a počítačové vidění** - porovnávání obrazů, předzpracování, klasifikace, stereo vidění, segmentace a další.
- ▶ **Zpracování signálů** - potlačení šumu (na telefonních linkách), analýza seismických signálů
- ▶ **Řízení** - řízení robotické paže, couvání auta s návěsem
- ▶ **Rozpoznávání vzorů** - ručně psaná číslce a písmena, klasifikace a analýza radarových dat
- ▶ **Medicína** - analýza signálu z EEG, EKG, diagnostikování nemocí
- ▶ **Armáda** - detekce podvodních min, klasifikace radarových obrazů

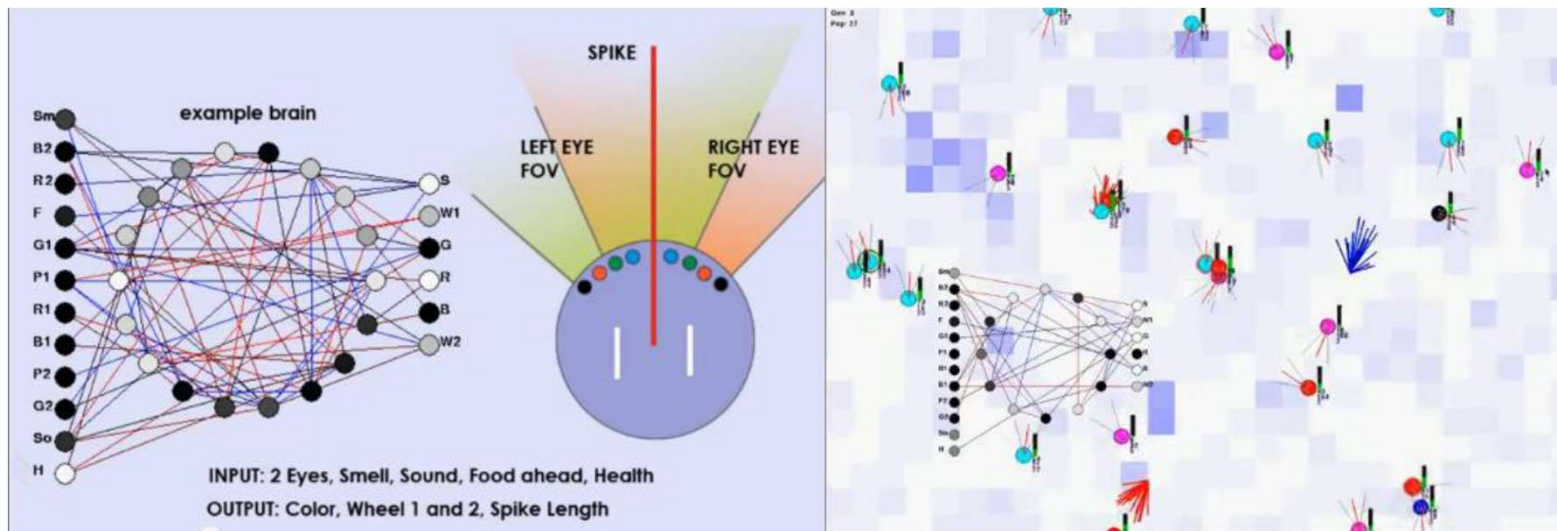
Aplikace umělých neuronových sítí

- ▶ **Generování a rozpoznávání řeči**
- ▶ **Ekonomika** - vyhodnocování půjček, předpovědi vývoje trhu
- ▶ **Energetika** - předpověď odběru, detekce poruch, odhad stavu systému
- ▶ **Optimalizace** - komplexní a nelineární optimalizační problémy
- ▶ **Aproximace funkcí** - modelování, náhrada složitých matematických funkcí

Příklady použití - detekce objektů



Simulating evolution using social bots with multilayer perceptron



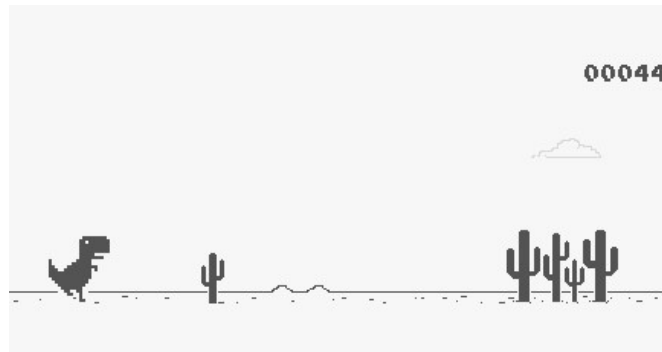
► <http://www.youtube.com/watch?v=GvEywP8t12I>

Mario-AI



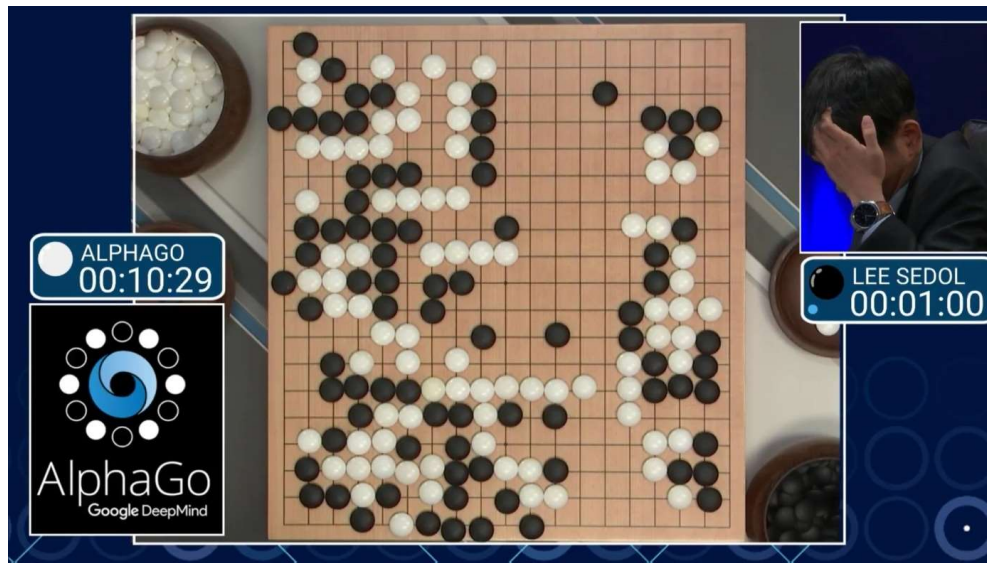
- ▶ Mario-AI championship 2012:
- ▶ <http://www.marioai.org/>
- ▶ <http://youtu.be/EsPcH31WEtc>
- ▶ <http://youtu.be/2vnPq5rijWE>
- ▶ <http://youtu.be/DlkMs4ZHhr8>
- ▶ <http://www.doc.ic.ac.uk/~rb1006/projects:marioai>

Google Chrome dinosaur



► https://youtu.be/sB_IgstiWlc

AlphaGo



- ▶ Hluboká neuronová síť
- ▶ Od roku 2015 poráží profesionální hráče
- ▶ <https://youtu.be/IFmj5M5Q5jg>



Základy MATLABu - Čísla

- ▶ cokoliv začíná číslicí, je považováno za číslo
- ▶ řádová čárka je zadána tečkou: 5.6; 565.84
- ▶ lze využít i symbol 'e' pro definici násobku 10x :
- ▶ $5e2 = 5 * 10^2 = 500$, $4.589e-7 = 0,0000004589$
- ▶ imaginární jednotka je zadána symbolem 'j' nebo 'i': $j = \sqrt{-1}$, $2 + 3j$, $3e - 5i$,

Popis základních příkazů

- ▶ `clear variables` - vymazání všech proměnných
- ▶ `close all` - uzavření všech grafických oken
- ▶ `clc` - vymazání příkazového okna
- ▶ `help elfun` - seznam elementárních funkcí
- ▶ `csvread('*.csv')` - načte soubor `*.csv`
- ▶ `csvwrite('*.csv', data)` - uloží do souboru `data`

Popis základních příkazů

- ▶ `save('*.mat', 'data')` - uloží do binárního souboru `*.mat` proměnnou `data`. Může uložit více proměnných, případně celý workspace.
- ▶ `load('*.mat')` - načte proměnnou/é ze souboru. Proměnné jsou pojmenovány stejně jako při ukládání do souboru.

Operátory

- ▶ Aritmetické
 - ▶ Maticové: +, -, *, /, ^, '
 - ▶ Vektorové - provádí se po jednotlivých elementech, před operátor se píše tečka: .*, ./, .^
- ▶ Relační: <, <=, >, >=, ==(je rovno), ~= (není rovno) - výsledek: nepravda (0) nebo pravda (1)
- ▶ Logické: & (and), | (or), ~ (negace)

Proměnné, vektory a matice

- ▶ Přiřazení hodnoty do proměnné: `prom = 100;`
- ▶ Vektor:
 - ▶ řádkový, oddělení čárkou nebo mezerou: `radek = [1, 2, 3];` `radek = [1 2 3];`
 - ▶ sloupcový, oddělení středníkem: `sloupec = [1;2;3]`
- ▶ Matice: `matice = [1 2 3;4 5 6;7 8 9];`

Práce s maticemi

- ▶ Indexování vektorů a matic začíná od 1
- ▶ Výběr prvku z matice:
 - ▶ jeden prvek (řádek, sloupec): `mat(2,3)`
 - ▶ podmatice: `mat(2:4, [3,5,7])` - prvky ze 2. až 4. řádku a z 3, 5 a 7 sloupce
 - ▶ všechny řádky (sloupce): `mat(: , 3)`
 - ▶ poslední číslo v řádku: `mat(5, end)`
 - ▶ matice jsou uloženy jako sloupce jdoucí za sebou, možnost indexovat pomocí jediného čísla: `mat(10)`

Indexování matic

		Sloupce (n)				
		1	2	3	4	5
Řádky (m)	1	1 ¹	2 ⁶	3 ¹¹	4 ¹⁶	5 ²¹
	2	6 ²	7 ⁷	8 ¹²	9 ¹⁷	10 ²²
	3	11 ³	12 ⁸	13 ¹³	14 ¹⁸	15 ²³
	4	16 ⁴	17 ⁹	18 ¹⁴	19 ¹⁹	20 ²⁴
	5	21 ⁵	22 ¹⁰	23 ¹⁵	24 ²⁰	25 ²⁶

M(3,1)
M(3)

M(4:5,2:3)
M([9 14;10 15])

M(1:5,5)
M(:,5)
M(21:25)
M(1:end,end)
M(:,end)
M(21:end)

Tvorba matic

- ▶ `zeros(r, s)` - nulová matice $r \times s$
- ▶ `ones(r, s)` - jedničková matice $r \times s$
- ▶ `eye(r, s)` - jednotková matice $r \times s$
- ▶ `rand(r, s)` - hodnoty v intervalu $(0,1)$ matice $r \times s$
- ▶ `od:krok:do` - vytvoření vektoru hodnot, bez uvedení kroku se bere roven 1

Funkce pracující s maticemi

- ▶ `[r, s] = size(mat)` - vrací počet řádků a sloupců
- ▶ `k = length(mat)` - u vektoru vrací počet prvků, u matice větší hodnotu z počtu řádků a sloupců
- ▶ `sum(mat)` - vrací řádkový vektor součtů prvků ve sloupcích
- ▶ `max(mat)` - vrací řádkový vektor maximálních prvků ve sloupcích
- ▶ `min(mat)` - vrací řádkový vektor minimálních prvků ve sloupcích

Větvení a cyklus

- ▶ *if logická podmínka*
 pokud je splněna
else
 pokud není splněna
end
- ▶ *for řídicí proměnná = pole hodnot*
 tělo cyklu
end



Vykreslování grafů

- ▶ `figure` - otevření nového okna grafu,
- ▶ `figure(2)` - otevření nebo přepnutí na okno grafu 2
- ▶ `subplot(k,l,m)` - rozdělení aktivního grafického okna na více grafů v k řádcích a l sloupcích, přepnutí do m.
- ▶ `plot(x,y)` - vykreslení spojitého 2D grafu
- ▶ `plot(x,y,'--y')` - vykreslení grafu čárkovaně a žlutou barvou
- ▶ `title('nazev')` - doplnění názvu grafu
- ▶ `xlabel('osa x'), ylabel('osa y')` - doplnění popisky osy x, y

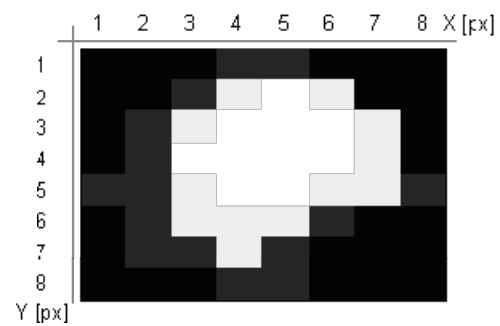
Vykreslování grafů

- ▶ `legend({'1. prubeh', '2. prubeh',...})` - doplnění legendy k jednotlivým průběhům v grafu
- ▶ `grid('on')`, `grid('off')` - zapnutí nebo vypnutí mřížky v grafu
- ▶ `hold('on')`, `hold('off')` - podržení nebo zrušení držení grafického výstupu; další graf bude zakreslen přes stávající
- ▶ `clf` - vymazání obsahu aktivního grafického okna

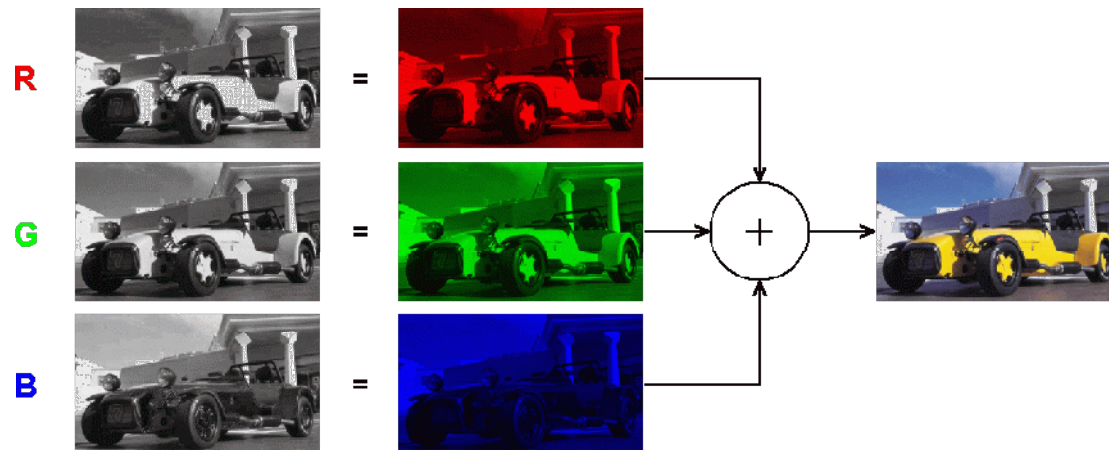
Skripty

- ▶ použití pro opětovné spuštění posloupnosti příkazů
- ▶ uloženy v textovém souboru s příponou *.m
- ▶ při spuštění jsou vyhledávány v aktuálním adresáři a poté ve vyhledávací cestě
- ▶ znak procenta uvozuje komentář až do konce řádku
- ▶ sekvence „procento procento“ vytvoří blok kódu spustitelný samostatně pomocí Ctrl+Enter
- ▶ pro přehlednost lze dlouhý řádek rozdělit na dva nebo více třemi tečkami na konci řádku ...
- ▶ pracují v hlavním pracovním prostoru - proměnné definované ve skriptu jsou viditelné i v ostatních skriptech a v příkazovém okně a obráceně

Reprezentace obrazu v počítači



	1	2	3	4	5	6	7	8	X [px]
1	0	0	0	10	10	0	0	0	
2	0	0	10	60	100	60	0	0	
3	0	10	60	100	250	100	60	0	
4	0	10	100	250	250	100	60	0	
5	10	10	60	100	100	60	60	10	
6	0	10	60	60	60	10	0	0	
7	0	10	10	60	10	0	0	0	
8	0	0	0	10	10	0	0	0	
Y [px]									



Zadání cvičení

Příklady vypracujte jako jeden skript v prostředí MATLAB



Příklad 1.

- ▶ Načtěte data ze souboru data1.csv, jedná se o dvourozměrnou matici.
 - ▶ Maticově vynásobte 1. řádek s transponovaným 2. řádkem
 - ▶ Maticově vynásobte transponovaný 1. řádek s 2. řádkem
- ▶ Zjistěte rozměry původní matice a obou výsledných matic

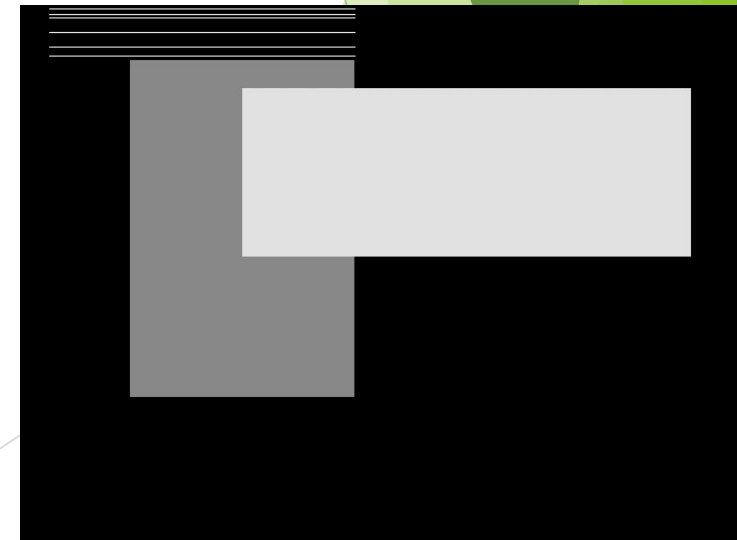
Příklad 2.

- ▶ Načtěte přiložené bitmapy (číslíce) do dvourozměrné matice (10 řádků a počet sloupců odpovídá počtu pixelů v obrázku) ve formátu jeden obrázek na jeden řádek.
- ▶ Následně vykreslete obrázky uložené v matici, viz ukázka.
- ▶ Řešení s více jak 10 řádky se neuznává.
- ▶ Užitečné příkazy:
 - ▶ Načtení bitmapy do matice:
`A = imread([num2str(i) '.bmp']);`
 - ▶ Vykreslení obrázku uloženého v matici:
`imshow(A);`
 - ▶ Změnění rozměrů matice:
`out = reshape(in, řádky, sloupce);`



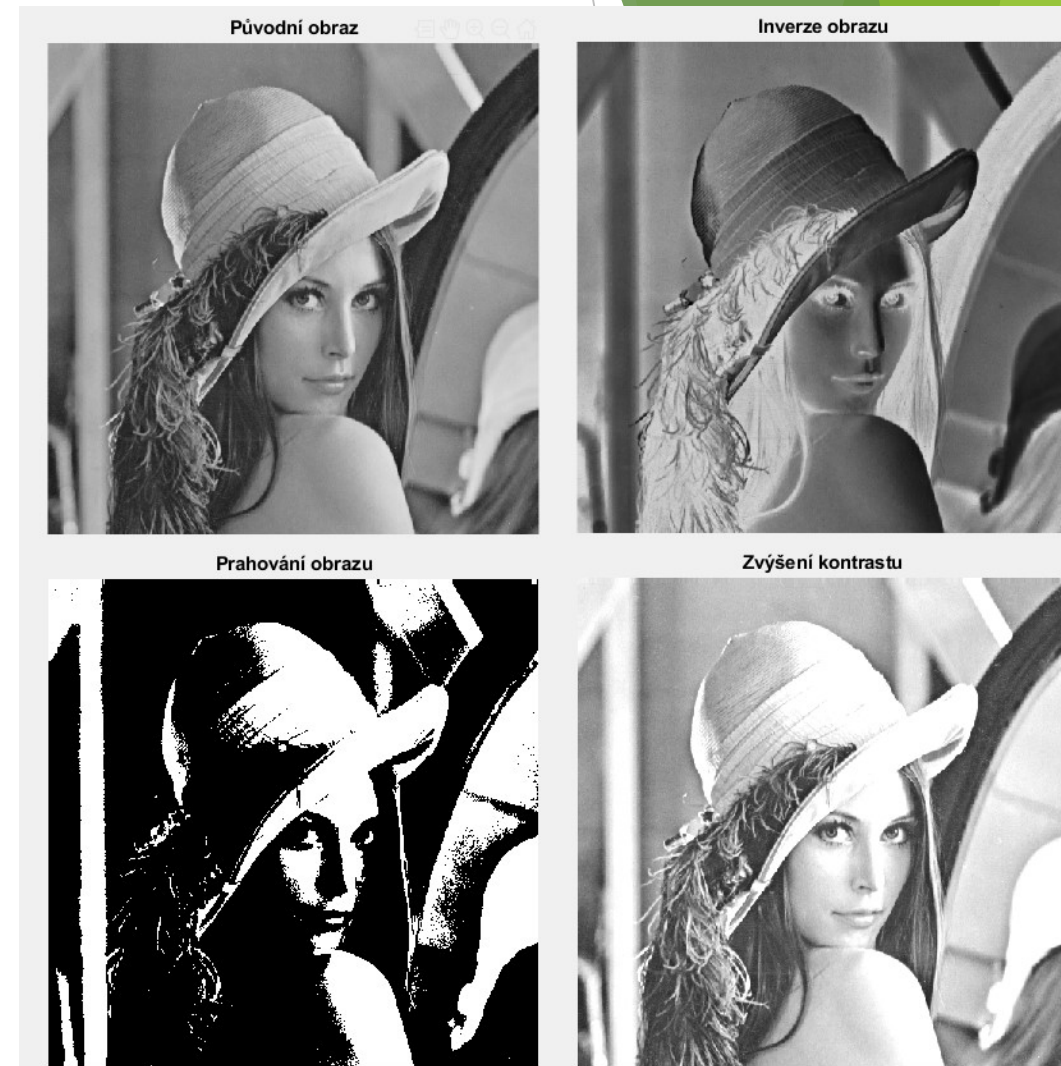
Příklad 3.

- ▶ Vytvořte matici šedotónového obrazu s rozměry 640x480px jako proměnnou typu uint8
- ▶ Vytvořte v obraze šedivé obdélníky
 - ▶ Pomocí funkce for, rozměry 200x300px, levý horní roh v bodě [100, 50], stupeň šedi 136
 - ▶ Pomocí nativního zápisu matlabu, rozměry 400x150px, počáteční bod [200, 75], stupeň šedi 225
- ▶ Řádek 4, 9, 12, 25, 38 a 46 a sloupce 28 až 300 nastavte na hodnotu 232
- ▶ Výsledný obraz zobrazte na rozsahu hodnot 0-255
- ▶ Proveďte součet prvků výsledné matice:
 - ▶ použití cyklů for
 - ▶ použití funkce sum()
 - ▶ bez použití +, - a sum(), např. násobením matic



Příklad 4.

- ▶ Načtěte obrázek lenaM.bmp
- ▶ Proveďte inverzi obrazu
- ▶ Proveďte prahování obrazu zvolenou hodnotou
- ▶ Zvyšte jas obrazu faktorem 1.2
- ▶ Přehledně v jednom okně zobrazte původní obraz a výsledky jednotlivých dílčích operací
- ▶ K jednotlivým obrazům doplňte popisky



Bonus příklad

- ▶ Načtěte soubor data2.csv, jedná se o dvou řádkovou matici.
- ▶ Načtěte prvky se sudým indexem z prvního řádku do jednoho vektoru a s lichým do druhého vektoru.
- ▶ To stejné proveďte i pro druhý řádek a další dva vektory.
- ▶ Vytvořte grafické okno se dvěma grafy pod sebou:
 - ▶ do prvního grafu vykreslete sudá data z obou řádků (různobarevně, prvních 100 hodnot)
 - ▶ do druhého vykreslete lichá data z obou řádků (různobarevně , prvních 100 hodnot)
 - ▶ oba grafy pojmenujte a udělejte popisky os

