



Politechnika  
Śląska

# WIZYJNY SYSTEM LOKALIZACJI I KLASYFIKACJI OBIEKTÓW DEDYKOWANY DO ROBOTA EKSPLORACYJNEGO

inż. Michał SOBCZYK

Promotor: dr hab. inż. Piotr PRZYSTAŁKA, prof. PŚ

WYDZIAŁ MECHANICZNY TECHNOLOGICZNY

KATEDRA PODSTAW KONSTRUKCJI MASZYN

Gliwice, 17.07.2020

---

# Plan prezentacji

- Geneza i cel pracy
- Projekt systemu
- Metodyka badań
- Badania weryfikacyjne
- Implementacja systemu
- Podsumowanie i wnioski

2

# Geneza i cel pracy

## Geneza

W związku z ciągłym rozwojem projektu Silesian Phoenix pojawiła się potrzeba automatyzacji wykonywania zadań przez robota w ramach zawodów ERC. W tym celu wykorzystać można system wizyjny.

## Cel pracy

Celem pracy jest opracowanie wizyjnego systemu lokalizacji i klasyfikacji obiektów dedykowanego dla robota eksploracyjnego.

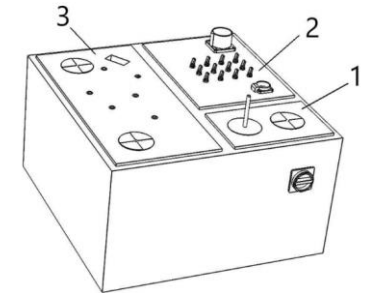


Rys. 1: Łazik marsjański [1]

# Projekt systemu

## – najważniejsze założenia

- Wykorzystanie wstępnie nauczonej sieci neuronowej oraz przetrenowanie z wykorzystaniem przygotowanych danych uczących
- Rozdzielczość zdjęć stanowiących zbiór uczący 608x608 px
- Proces uczenia przeprowadzony na komputerze PC (Intel Core i7-3930K, 48 GB RAM, 512 GB SSD, system Windows 10) z wykorzystaniem GPU (RTX 2070)
- Implementacja systemu na układzie Jetson Nano (system Linux w wersji Debian R32.3.1)
- Rozpoznawanie elementów wymienionych w regulaminie ERC 2020
- Praca systemu w czasie rzeczywistym
- Oczekiwana średnia precyzja sieci 90 %



Rys. 2: Wstępny projekt panelu [2]

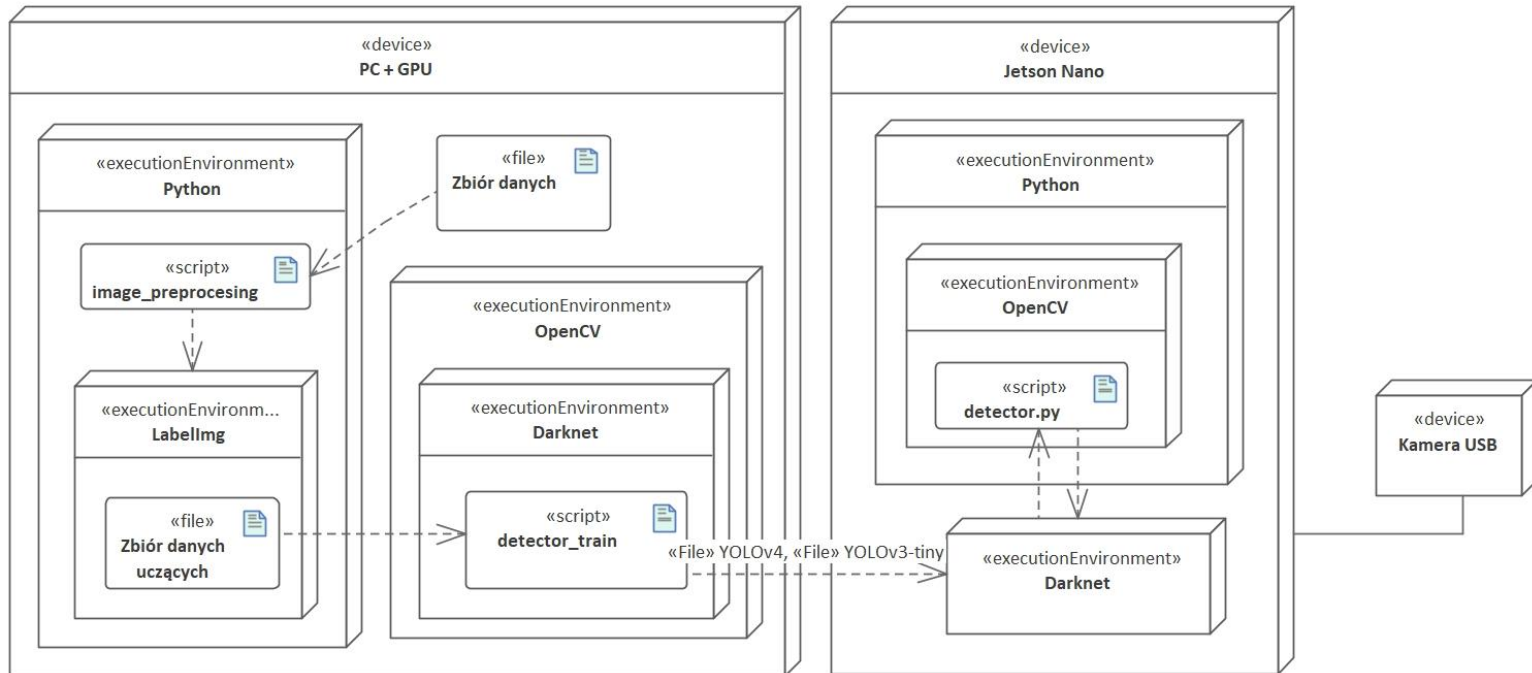


Rys. 3: Zdjęcie panelu z zawodów ERC [1]

# Projekt systemu

– diagram wdrożeniowy

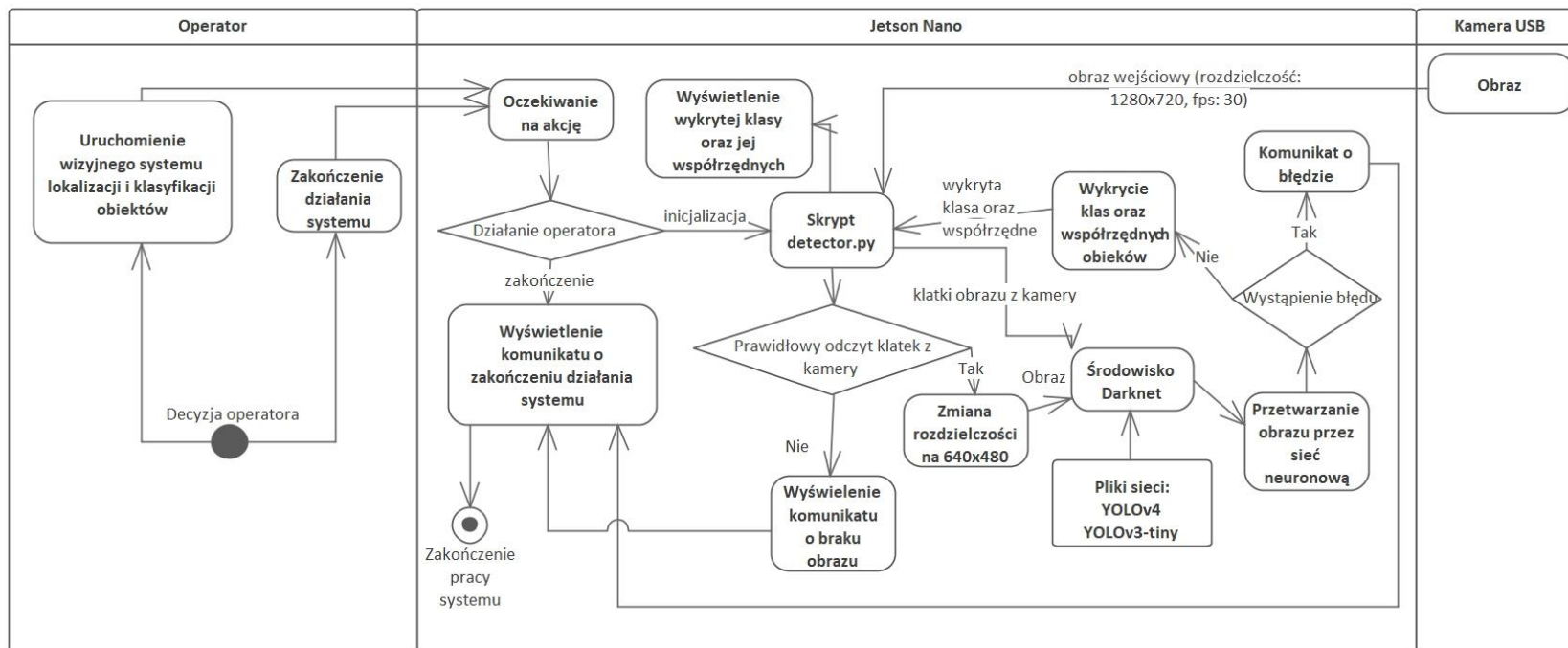
5



Rys. 4: Diagram wdrożeniowy systemu

# Projekt systemu

## – diagram czynności



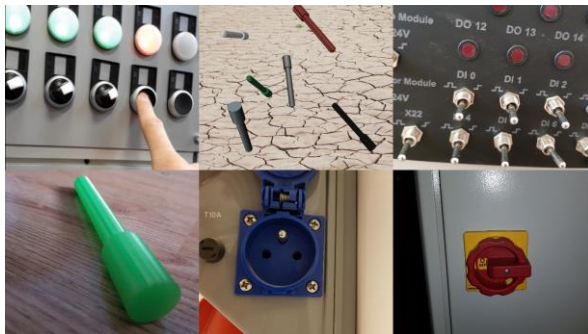
Rys. 5: Diagram czynności systemu

6

# Metodyka badań

– zbiór danych uczących

7



Rys. 6: Przykładowe zdjęcia wchodzące w skład zbioru uczącego



Rys. 7: Nałożone ramki określające położenie obiektu

Tab. 1: Nazwa oraz ilość utworzonych etykiet

Nazwa elementu	Nazwa etykiety	Liczba zdjęć
próbka	sample	101
gniazdo elektryczne	socket	248
przełącznik obrotowy	selector-switch	119
przełącznik krzywkowy	cam-switch	162
przełącznik dźwigniowy	toggle-switch	104

# Metodyka badań

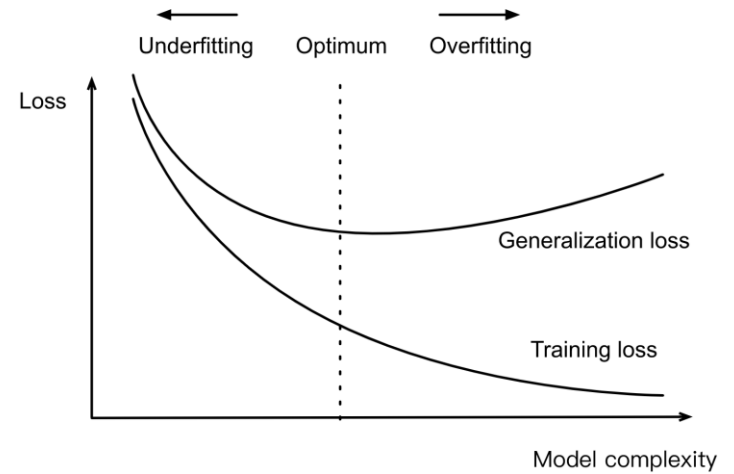
## – walidacja krzyżowa

8

4-krotna walidacja krzyżowa				
Eksperyment 1	█	█	█	█
Eksperyment 2	█	█	█	█
Eksperyment 3	█	█	█	█
Eksperyment 4	█	█	█	█

Zbiór walidacyjny  
 Zbiór trenujący  
 Zbiór testowy

Rys. 8: 4-krotna walidacja krzyżowa



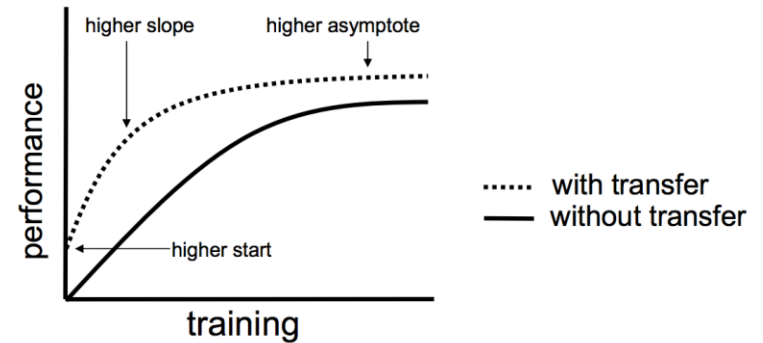
Rys. 9: Zjawisko nadmiernego dopasowania sieci do zbioru trenującego [3]



# Metodyka badań

– transfer wiedzy

- YOLOV4:
  - 137 konwolucyjnych warst
  - trenowana na zbiorze MS COCO [4]
- YOLOV3-TINY:
  - 15 konwolucyjnych warst
  - trenowana na zbiorze COCO trainval [5]



Rys. 10: Porównanie trenowania sieci z wykorzystaniem transferu wiedzy oraz bez [6]

# Badania weryfikacyjne

– plan badań

- Trenowanie sieci YOLOv4 oraz YOLOv3-tiny z wykorzystaniem 4-krotnej walidacji krzyżowej
- Przeprowadzenie ewaluacji na zbiorach testowych, obliczenie mAP, IoU, wyznaczenie macierzy pomyłek
- Wyliczenie średniej precyzji dla obu sieci
- Analiza i porównanie wyników
- Opracowanie wniosków

10

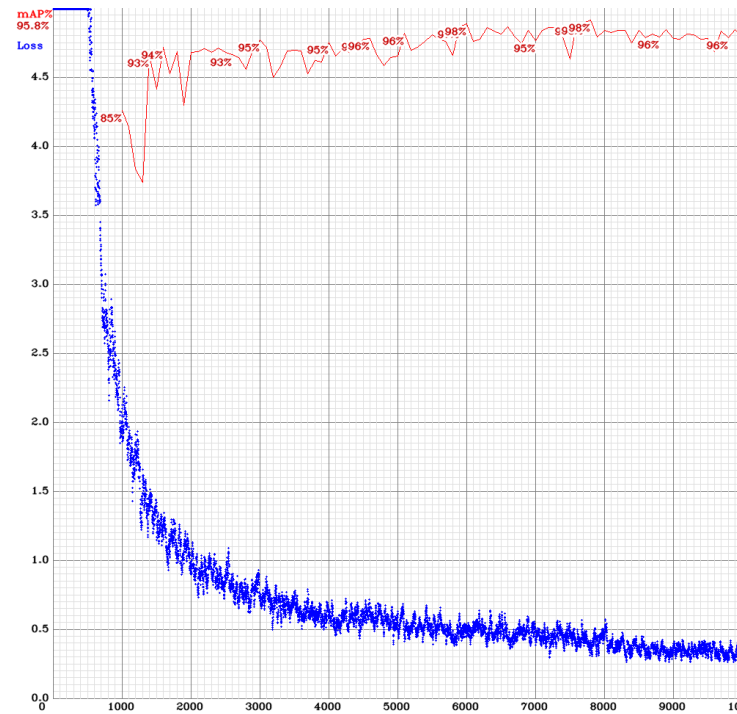


# Badania weryfikacyjne

## – trenowanie sieci neuronowej

Tab. 2: Dobrane parametry sieci

Parametr	Wartość
batch	64
subdivision	64
max-batches	10000
steps	8000,9000
width, height	608x608
classes	5
filters	30
mixup	1
cutmix	1
mosaic	1



Rys. 11: Wykres przebiegu trenowania-eksperyment 1, mAP oraz średnia strata – sieć YOLOv4

# Badania weryfikacyjne

– wybrane wyniki

Tab. 3: Uzyskane wyniki podczas walidacji krzyżowej sieci YOLOv4 oraz YOLOv3-tiny

Parametr	Wartość średnia uzyskana z 4-krotnej walidacji krzyżowej	
	YOLOv4	YOLOv3-tiny
mAP(t=0,5)	94,66%	73,66%
aloU	81,97%	70,5%
Precision	94,25%	92,75%
Recall	93,5%	64,75%
F1-score	93,75%	75,5%
TP	198	190
FP	17	38
FN	19	98
FPS (GPU: RTX 2070)	32	162
FPS (Jetson Nano)	1,1	9,4

Tab. 4: Uzyskane wyniki podczas walidacji krzyżowej poszczególnych klas z użyciem sieci YOLOv4

Element	Parametr	Nr. eksperymentu				
		1	2	3	4	średnia
sample	ap	99,15%	100%	99,71%	100%	99,72%
	TP	26	37	41	63	-
	FP	1	0	1	2	-
socket	ap	92,26%	85,40%	97,07%	91,57%	91,58%
	TP	79	76	87	81	-
	FP	5	5	3	14	-
toggle_switch	ap	95,99%	89,13%	95,10%	92,64%	93,215%
	TP	80	68	68	64	-
	FP	1	6	5	7	-
cam-switch	ap	100%	99,31%	100%	87,50%	96,70%
	TP	29	29	28	28	-
	FP	0	2	0	1	-
selector-switch	ap	91,41%	88,44%	95,59%	92,84%	92,07%
	TP	39	54	74	42	-
	FP	4	3	4	4	-

12

# Implementacja systemu

## – Jetson Nano

Wymagania wstępne:

- JetPack 4.3 (Linux debian R32.3.1)
- Python 3.7
- CUDA 10.0
- CUDNN 7.6.4
- OpenCV 4.2
- Darknet
- Numpy 1.18.4



Rys. 12: Jetson Nano Developer Kit



Rys. 13: Kamera USB

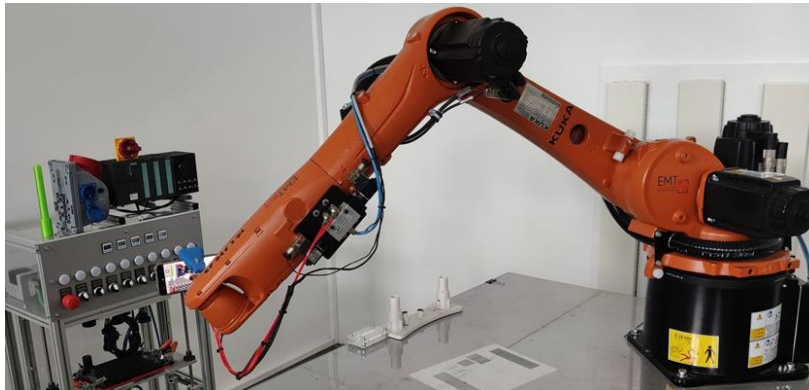
# Implementacja systemu

## – test systemu w warunkach rzeczywistych

Tab. 5: Wyniki testu w warunkach rzeczywistych sieci YOLOv4

Lp.	Parametr			Element				
	Dystans	Kąt	Światło	sample	socket	toggle-switch	cam-switch	selector-switch
1	200 cm	0°	100%	0	0	0	1	0,8
2	200 cm	0°	50%	0	0	0	1	1
3	200 cm	45°	100%	0	0	0	1	1
4	200 cm	45°	50%	0	0	0	1	1
5	100 cm	0°	100%	1	1	0,5	1	1
6	100 cm	0°	50%	1	1	0,5	1	1
7	100 cm	45°	100%	1	1	0,1	1	1
8	100 cm	45°	50%	1	1	0,2	1	1

14



Rys. 14: Scena przeprowadzanych badań



Rys. 15: Obraz wynikowy systemu wizyjnego

# Podsumowanie i wnioski

- Cel pracy został osiągnięty, opracowany system wizyjny spełnia określone założenia przy zastosowaniu sieci YOLOv4.
- Największy wpływ na skuteczność sieci ma zbiór danych uczących. Im większe podobieństwo elementów zbioru do warunków rzeczywistych pracy, tym większa precyzja działania systemu.
- Większa liczba konwolucyjnych warstw sieci zapewnia większą precyzję, jednak ze względu na bardziej skomplikowaną architekturę wymaga większej mocy obliczeniowej, w konsekwencji czego sieć osiąga mniejszą liczbę klatek na sekundę.
- Układ Jetson Nano dysponuje mocą obliczeniową pozwalającą na prace systemów wizyjnych opierających się o sieci neuronowe.
- Wykorzystanie transferu wiedzy pozwoliło na otrzymanie wysokiej precyzji działania sieci przy ograniczonym czasie trenowania.
- Zastosowanie augmentacji danych przy ograniczonym zbiorze uczącym zwiększa precyzję działania sieci.
- System zbudowany jest w sposób pozwalający na implementację na robocie eksploracyjnym jako część odpowiedzialna za lokalizację oraz detekcję obiektów.

15



# Wizyjny system lokalizacji i klasyfikacji obiektów dedykowany do robota eksploracyjnego



## Geneza.

W związku z ciągłym rozwojem projektu **Silesian Phoenix** pojawiła się potrzeba automatyzacji wykonywania zadań przez robota w ramach zawodów **ERC**. W tym celu wykorzystać można system wizyjny.



## Cel pracy.

Celem pracy jest opracowanie wizyjnego systemu lokalizacji i klasyfikacji obiektów dedykowanego dla robota eksploracyjnego.



## Wykrywane elementy.



Gniazdo



Próbka



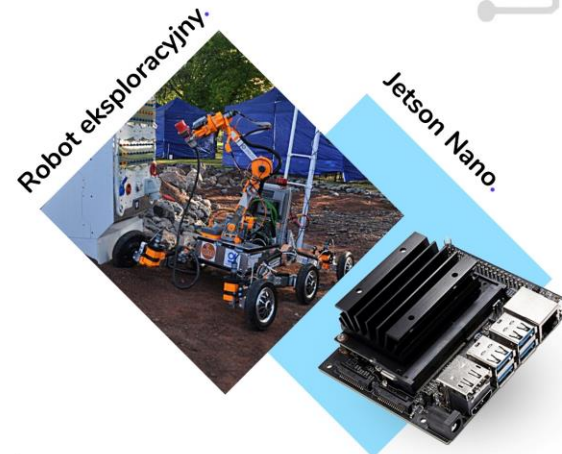
Przełącznik dźwigniowy



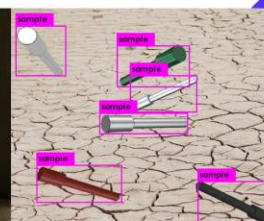
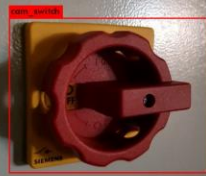
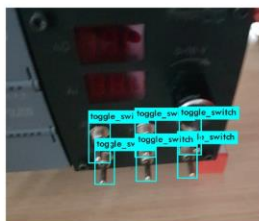
Przełącznik obrotowy



Przełącznik krzywkowy



## Przykładowe obrazy wyjściowe systemu:



## Wykorzystane narzędzia:

- Python 3.7
- Labellmg 1. 8. 0
- OpenCV 4. 1. 1
- Darknet
- YOLOv4, YOLOv3-tiny

Nazwa instytutu: Katedra Podstaw Konstrukcji Maszyn  
 Rodzaj pracy: Praca Dyplomowa Magisterska  
 Tytuł: Wizyjny system lokalizacji i klasyfikacji obiektów dedykowany do robota eksploracyjnego.  
 Autor: inż. Michał SOBCZYK  
 Kierunek: Mechatronika  
 Specjalizacja: ME4  
 Promotor: dr hab. inż. Piotr PRZYSTAŁKA, prof. PS  
 Rok Akademicki: 2019/2020



# Bibliografia

1. Strona Studenckiego Koła Naukowego Zastosowań Metod Sztucznej Inteligencji AI-METH <https://sknaimeth.polsl.pl/lazik-marsjanski/#jp-carousel-2659> dostęp: 10.02.2020
2. Regulamin zawodów ERC 2020
3. Dive into deep learning - [https://d2l.ai/chapter\\_multilayer-perceptrons/underfit-overfit.html](https://d2l.ai/chapter_multilayer-perceptrons/underfit-overfit.html)
4. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.
5. Pengyi Zhang and Yunxin Zhong and Xiaoqiong Li. SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications, 2019
6. J. Brownlee, Transfer learning for deep learning <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>

17

