# Zestawienie rozwiązań do wykrywania przeszkód

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rozwiązanie | Zastosowane metody | Skuteczność | Open Source | Dane | Liczba cytowań | GPU |
| StixelNet 2015 | Conditional Random Fields(CRF); graph-cut; podział obrazu na pionowe paski | F = 89.12%  AP=81.23% | <https://github.com/erlikn/stixelnet>  <https://github.com/dhernandez0/stixels> | KITTI | 41 | I tak i nie |
| SqueezeDet 2016 | Region Proposal Network(RPN); end-to-end training | AP=76.7% | <https://github.com/BichenWuUCB/squeezeDet>  <https://github.com/omni-us/squeezedet-keras> | KITTI | 47 | tak |
| MultiNet - 8 maj 2018 | CRF; end-to-end training | F = 94.88%  AP=93.71% | <https://github.com/MarvinTeichmann/MultiNet> | KITTI | 1 | tak |
| Faster - rcnn | Region Proposal Network; Translation-Invariant Anchors | AP=60% | <https://github.com/ShaoqingRen/faster_rcnn> |  | 30 | tak |
| Fcn – berkeleyvision 2015 | FCN-8s | Pixel acc. = 90.3% | <https://github.com/shelhamer/fcn.berkeleyvision.org> |  | 5272 | tak |
| Cn24 2015 | Incorporating spatial priors; | F = 86.5% | <https://github.com/cvjena/cn24> | KITTI | 55 | tak |
| Detectron – 24 sty 2018 | Rozwiązanie stosowane do wykrywania przeszkód oraz jako DeepPose ; korzysta z kilku rodzajów sieci np.: fast-rcnn , RPN, RetinaNet, ResNet | APmax=60% | <https://github.com/facebookresearch/Detectron> | COCO | 1 | tak |

AP – average precision – średnia precyzja

TP – true positive – dobra detekcja

FP – false positive – zła detekcja

FN – false negative – powinno wykryć, a nie wykrywa przeszkody w danym pikselu

TN – true negative – stwierdzenie braku przeszkody w odpowiednim miejscu(pikselu)

# Inne rozwiązania

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rozwiązanie | Zastosowane metody | Skuteczność | Open Source | Dane | Liczba cytowań | GPU |
| YOLO 2016 | https://www.cv-foundation.org/openaccess/content\_cvpr\_2016/papers/Redmon\_You\_Only\_Look\_CVPR\_2016\_paper.pdf | AP=63.4% | https://github.com/llSourcell/YOLO\_Object\_Detection | COCO | 1861 | tak |
| DeepLab | https://arxiv.org/pdf/1606.00915.pdf |  | https://bitbucket.org/aquariusjay/deeplab-public-ver2.git | COCO, PASCAL | 1669 | tak |
| Reinspect | https://www.cv-foundation.org/openaccess/content\_cvpr\_2016/papers/Stewart\_End-To-End\_People\_Detection\_CVPR\_2016\_paper.pdf | 76% | <https://github.com/russell91/tensorbox> | KITTI | 111 | nie |