

Használati útmutató a Carcounter program elindításához Windows 10 felhasználóknak

1. Visual Studio telepítése

A Visual Studio fejlesztői környezetben nyílik lehetőség a forráskódok megírására és tesztelésére. A Microsoft honlapjáról lehet letölteni a legújabb változatot (Visual Studio 2019, link: <https://visualstudio.microsoft.com/downloads/>). Az én gépemem előre fel volt telepítve a 2017-es verzió én így azt használtam a projekt során.

2. OpenCV telepítése

Az OpenCV egy nyílt forráskódú szoftver könyvtár, ami a gépi látásnál és tanulásnál használatos algoritmusokat és függvényeket tartalmazza. Az OpenCV oldalán (link: <https://opencv.org/releases/>) számos verziója elérhető (jelenleg a legújabb a 4.3.0), a projekthez legalább a 3.4.2-es verziót kell letölteni (a YOLOv3 algoritmust innentől támogatja az OpenCV). Én a projekt során a 3.4.9-es verziót használtam. Az OpenCV telepítésének lépései:

- I. Az <https://opencv.org/releases/> oldalon válasszuk ki a nekünk szimpatikus, legalább 3.4.2-es verziójú OpenCV telepítőjét. (Windows). Ezután automatikusan elindul a letöltés.
- II. Készítsünk el a C: meghajtóra egy "C:\OpenCV-X.X.X" nevű mappát, ahol X.X.X jelöli a verziószámot (pl.: "C:\OpenCV-3.4.9"). A letöltött telepítő fájlt indítsuk el, az "Extract to" opción belül válasszuk ki az imént létrehozott mappa elérési útvonalát.
- III. A telepítés befejezése után adjuk hozzá az általunk feltelepített verziójú Visual Studiohoz és OpenCV-hez tartozó *bin* directory-t az operációs rendszer Path-jához. A hozzáadandó *bin* elérési útvonalának formátuma a következő: "C:\OpenCV-X.X.X\opencv\build\x64\vcYY\bin", ahol az X.X.X ismét az OpenCV verzióját jelöli, az vcYY értéke pedig a használt Visual Studio változatától függ (2015-ös verzió esetén vc14, 2017: vc15, 2019: vc16). Például OpenCV 3.4.9 és Visual Studio 2017 esetén "C:\OpenCV-3.4.9\opencv\build\x64\vc15\bin".
(Elérési útvonal hozzáadása a Path-hoz Windows 10-ben: Vezérlőpult → Rendszer → Speciális rendszerbeállítások → Környezeti változók → Rendszerváltozók ablakon belül a *Path* sorra kattintunk → Szerkesztés → Új → Beírjuk a *bin* elérési útvonalát, majd OK-zunk mindent.)

3. A programfájlok letöltése

A programfájlokat a <https://github.com/MartonPolcz/Cpp-CarCounting-with-YOLOv3> oldalról lehet letölteni. A szükséges programfájlok a következők: **BoundingBox.h**, **BoundingBox.cpp**, **Carcounter.cpp**. A bemeneti videó fájl a **traffic.mp4**. A YOLOv3 algoritmushoz szükséges neurális háló konfigurációs-, súly- és névfájljai a következő helyeken érhetőek el:

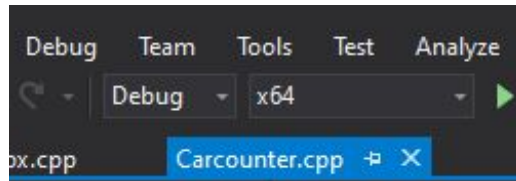
- **coco.names**: <https://github.com/pjreddie/darknet/blob/master/data/coco.names>
- **yolov3-tiny.cfg**: <https://github.com/pjreddie/darknet/blob/master/cfg/yolov3-tiny.cfg>
- **yolov3-tiny.weights**: <https://github.com/smarthomefans/darknet-test/blob/master/yolov3-tiny.weights>

4. A Visual Studio elindítása, új projekt létrehozása, beállítások elvégzése

Indítsuk el a Visual Studiot. A bal felső sarokban File → New → Project. A felugró ablak bal oldali sávjában válasszuk ki a *Visual C++* mezőt, válasszuk ki az *Empty Project*-et, adjunk nevet a projektnek (pl.: Carcounter), válasszuk ki a projekt helyét (pl.: Asztal), és gondoskodjunk arról, hogy a jobb alsó sarokban található "Create directory for solution" és "Add to Source Control" boxok ne legyenek bejelölve. Ezután nyomjunk OK-t, így létrejön a projektünk. A jobb oldali *Solution Explorer* ablakban jobb klikk a *Header Files*-ra, Add → New Item, válasszuk ki a *Header file* mezőt, névként adjuk meg hogy *BoundingBox.h* és kattintsunk az Add-ra. Ugyanezt ismételjük meg még kétszer, csak ezúttal *Source Files*-ra jobb klikk, a *C++ file* mezőt válasszuk ki, névként pedig egyszer *BoundingBox.cpp*, másodszor pedig *Carcounter.cpp*-t adjunk meg. Az

így létrehozott header és cpp fájlokba másoljuk be a **BoundingBox.h**, **BoundingBox.cpp** és a **Carcounter.cpp**-ben található kódokat, majd mentjük el a projektet.

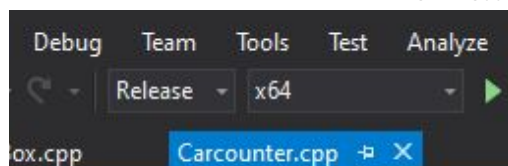
Hogyha ez kész van, győződjünk meg arról, hogy az ablak tetején található "Solution Configurations" gördülő menüje **Debug**-ra, a "Solution Platforms" menüje pedig **x64**-re van állítva.



Ezután a Visual Studioban a következő beállításokat hajtjuk végre:

1. Felül *Project* fül → (projekt neve) *Properties* → *Configuration properties* → *VC++ directories* → *Include directories*, és itt adjuk hozzá az OpenCV verziókhöz tartozó include directory-t (pl.: `C:\OpenCV-3.4.9\opencv\build\include;$(IncludePath)`).
2. Felül *Project* fül → (projekt neve) *Properties* → *Configuration properties* → *VC++ directories* → *Library directories*, és itt adjuk hozzá az OpenCV és Visual Studio verziókhöz tartozó include directory-t (pl.: `C:\OpenCV-3.4.9\opencv\build\x64\vc15\lib;$(LibraryPath)`).
3. Felül *Project* fül → (projekt neve) *Properties* → *Configuration properties* → *Linker* → *Input* → *Additional dependencies*, és itt adjuk hozzá az OpenCV és Visual Studio verziókhöz tartozó debug libraryket (d-vel végződő .lib fájlokat). OpenCV 3.4.9 esetén és általánosságban csak egy debug library szokott lenni, ez pedig az `opencv_worldXXXd.lib`, ahol XXX ismételtlen a verziószámot jelöli. Ezt a fájlnevet adjuk hozzá az *Additional dependencies*-hez.
4. Felül *Project* fül → (projekt neve) *Properties* → *Configuration properties* → *Debugging* → *Command Arguments*, és ide másoljuk be a következőt: `./object_detection_yolo.out -video=traffic.mp4`. Ekkor a `traffic.mp4` videóra fog lefutni az autószámláló program, amire azt eredetileg is optimalizáltuk. Hogyha más bemeneti videó fájlt szeretnénk megadni, akkor egyszerűen lecseréljük arra a `traffic.mp4`-et.

Hogyha gyorsabban szeretnénk a programunkat lefuttatni, akkor a "Solution Configurations" menüt állítsuk **Release**-re, a "Solution Platforms" menüt pedig hagyjuk **x64**-re van állítva.



Ezután végezzük el a fenti **1-4. lépéseket** újra, azzal a lényeges különbséggel, hogy a 3. lépésben `opencv_worldXXXd.lib` helyett `opencv_worldXXX.lib`-et adunk meg.

5. A letöltött fájlok hozzáadása a projekthez, a program futtatása

Utolsó lépésként a letöltött **traffic.mp4**, **coco.names**, **yolov3-tiny.cfg** és **yolov3-tiny.weights** fájlokat másoljuk be a projekt mappájába. Ezután az **F5** gombot megnyomva debuggolva, az **F5+Ctrl** billentyűkombinációt megnyomva pedig debuggolás nélkül futtathatjuk le a programunkat. A program lefutása után a képfeldolgozási időket tartalmazó szövegfájl a projekt mappában `times.txt`, a kimeneti videó fájl pedig `traffic_yolo.avi` néven lesz elérhető.