TASK MP.7:

count the number of keypoints on the preceding vehicle for all 10 images and take note of the distribution of their neighborhood size. Do this for all the detectors you have implemented.

Answer: According to the results below the best choice for the maximum numbers of keypoints would be **1-BRISK 2-AKAZE 3-FAST**

	SHITOMASI	**HARRIS**	**FAST**	**BRISK**	**ORB**	**AKAZE**	**SIFT**
Img 0	125	17	149	264	92	166	138
Img 1	118	14	152	282	102	157	132
Img 2	123	18	150	282	106	161	124
Img 3	120	21	155	277	113	155	137
Img 4	120	26	149	297	109	163	134
Img 5	113	43	149	279	125	164	140
Img 6	114	18	156	289	130	173	137
Img 7	123	31	150	272	129	175	148
Img 8	111	26	138	266	127	177	159
Img 9	112	34	143	254	128	179	137

TASK MP.8:

count the number of matched keypoints for all 10 images using all possible combinations of detectors and descriptors. In the matching step, use the BF approach with the descriptor distance ratio set to 0.8.

Answer: According to the results below the best choice for the maximum number of matched keypoints would be **1-BRISK+SIFT 2- BRISK+BRIEF 3- BRISK+BRISK**

Detector Descriptor	**SHITOMASI**	**HARRIS**	**FAST**	**BRISK**	**ORB**	**AKAZE**	**SIFT**
BRISK	690	121	776	1298	649	1110	536
BRIEF	816	141	883	1344	450	1087	597
ORB	768	145	862	933	530	918	Out Of Memory
FREAK	574	123	667	1093	346	973	506
AKAZE	-	-	-	-	-	1172	-
SIFT	927	163	1046	1646	763	1270	800

TASK MP.9:

log the time it takes for keypoint detection and descriptor extraction. The results must be entered into a spreadsheet.

Detector / Descriptor	**SHITOMASI**	**HARRIS**	**FAST**	**BRISK**	**ORB**	**AKAZE**	**SIFT**
BRISK	3.048	2.943	2.675	6.255	2.727	4.668	4.366
BRIEF	0.243	0.211	0.068	3.603	0.160	2.158	1.548
ORB	0.329	0.236	0.069	3.443	0.280	2.120	-
FREAK	0.401	0.384	0.317	3.543	0.422	2.114	2.066
AKAZE	-	-	-	-	-	3.572	-
SIFT	0.402	0.399	0.387	4.031	1.013	2.328	2.512

TOP3 detector/description extractor are:

- **FAST+SIFT (1046 MkeyPoints, 0.387 sec)**
- **AKAZE+FREAK (1087 MkeyPoints, 2.158 sec)**
- **FAST+BRIEF (883 MkeyPoints, 0.068 sec)**

I preferred to combine between looking for the max. keypoints matched and the fast run time. That is why I avoided choosing BRISK-BRISK or BRISK_SIFT for example.