Project Page for OpenCV Google Summer of Code 2017 (GSoC 2017)

General Information:

We are in!

- Program Site for GSoC 2017
  - General GSoC 2017 site
  - OpenCV's GSoC 2017 organization/acceptance page
- Mailing list for OpenCV GSOC 2017: `opencv-gsoc-2017@googlegroups.com`
  - Group site https://groups.google.com/forum/#!forum/opencv-gsoc-2017
- IRC Channel: `#opencv` on freenode
- **Timelines**
  - Timeline for GSoC 2017
  - Major GSoC 2017 Milestones

Important dates:

Date	Description	Comment
Oct 10, 2016	Program announced	
Jan 19, 16:00 UTC	Mentoring organizations begin submitting appls to Google	**DONE**
Feb 9, 16:00 UTC	Mentoring organization application deadline	**DONE**
Feb 10 - 26	Google program administrators review organization applications	**DONE**
Feb 27, 16:00 UTC	List of accepted mentoring organizations published	WE'RE IN!
Feb 27 - Mar 20	Potential student participants discuss application ideas with mentoring organizations	Send proposals to GSoC site!
Mar 20, 16:00 UTC	Student application period opens	Send proposals to GSoC site!
Apr 3, 16:00 UTC	Student application deadline	**DONE!** That is, unless you have your own $, too late now
Apr 19, 16:00 UTC	Slots awarded	**Thanks Google!**
May 4, 16:00 UTC	Accepted student proposals announced	**Rev your engines as of 16:00 UTC!**
Community Bonding Period	Students get to know mentors, read documentation, get up to speed to begin working on their projects	**Bonded**

May 30	Coding officially begins!	
Work Period	Students work on their project with guidance from Mentors	
Jun 26, 16:00 UTC	Mentors and students can begin submitting Phase 1 evaluations	
Jun 30, 16:00 UTC	Phase 1 Evaluation deadline; Google begins issuing student payments	
Work Period	Students work on their project with guidance from Mentors	
Jul 24, 16:00 UTC	Mentors and students can begin submitting Phase 2 evaluations	
Work Period	Students continue working on their project with guidance from Mentors	
Jul 28, 16:00 UTC	Phase 2 Evaluation deadline	
Aug 21 - 29, 16:00 UTC	Final week: Students submit their final work product and their final mentor evaluation	:checkered_flag:
Aug 29 - Sep 5, 16:00 UTC	Mentors submit final student evaluations	
Sep 6	Final results of Google Summer of Code 2017 announced	:tada:
Late October	Mentor Summit at Google	

Times:

UTC to PDT (California uses PST in the winter (from Nov 1st) and PDT in the summer (from March 8)).

UTC time

UTC time converter

Resources:

- OpenCV official Site
- OpenCV wiki
- [[How to do a pull request/How to Contribute Code|How_to_contribute]]
- Source Code can be found at GitHub/opencv and GitHub/opencv_contrib
- [[Developer meeting notes|Meeting_notes]]

How you will be evaluated if you are an accepted student

- Student projects to be paid only if:
  - **Midterm 1:**
    - You must generate a pull request
      - That builds
      - Has at least stubbed out functionality

        - With appropriate Doxygen documentation
        - Has at least stubbed out unit test
        - Has a stubbed out example of use that builds

- **Midterm 2:**
    - You must generate a pull request
        - That builds
        - Has basic functionality
        - With appropriate Doxygen documentation
        - Has basic unit test

- **End of summer:**
    - A full pull request
        - Full Doxygen documentation
        - A good unit test
        - Example of use code
    - Create a (short!) Movie (preferably on Youtube, but any movie) that demonstrates your code
        - We use this to create an overall summary. Past years:
            - The 2015 Movie
            - The 2014 Movie
            - The 2013 Movie

For students interested in applying

1. You **must** already know how to program fluently in C++
    - Some projects may instead specifically require Python or Matlab skills
2. Ask to join the OpenCV GSoC Forum List
    - Discuss projects ideas below or your own ideas with OpenCV mentors on the list now and April.
    - Always title your proposal with what you want to do (example: *Implement Patch Match Stereo Algorithm*)
        - **NOTE:** The above is to discuss proposals with mentors. **BUT**, when the application period starts, you must still sign up with Google Summmer of Code and submit your proposal to the OpenCV organization. If not, you will not show up on the database where we can select you as a student.
3. In March, Go to the GSoC site and sign up to be a student with OpenCV
4. Post the project from below or your own agreed on project on the GSoC to opencv-gsoc-2017@googlegroups.com
    - Include Name, google email, age
    - Include how you think you are qualified to accomplish this project (skills, courses, relevant background)
    - Include Country of origin, school you are enrolled in, Professor you work with (if any)
    - Include a projected timeline and milestones for the project
5. Once (and if!) OpenCV gets accepted as GSoC org this year, and we are told how many slots we will get **and** you've signed up for a project with us in March before the April 3rd deadline: **Then:**
    - We will weigh the students and projects against the mentors we gather and the mentor's interests and choose which students/project to pursue.
    - Accepted students will be posted on the GSoC site in May (and we will notify the accepted students ourselves).

- Students are paid over the summer by Google **IF** the mentor accepts the student's work. There are several milestone based go-nogo points.

For computer vision professionals interested in mentoring

1. Contact us on the opencv-gsoc googlegroups mailing list above and ask to be a mentor (or we will ask you in some known cases)
2. If we accept you, we will post a request from the Google Summer of Code OpenCV project site asking you to join.
3. You must accept the request and **you are a mentor!**
4. You then:
   - Go to the opencv-gsoc googlegroups mailing list above and look through student project proposals, find a student and project you like and work with them to refine a realistic proposal that they can implement in a summer (you have to judge whether the student is capable - **absolutely no non-coders in the language you need, typically C++, accepted!** Summer is too short to learn to code and get something done).
     - you may also find (good) students and get them to apply, perhaps to your pet project idea
   - Once you find or create a project proposal that you want to mentor
     - several students might try for the same project
     - alternatively, you might have to convince a student to change projects to one you like or recruit an external student to join Google Summer of Code and apply to your project
   - But, always encourage students to officially apply through the Google Summer of Code site - it helps us and them.
5. We later get a slot allocation from Google, the administrators then *"spend"* the slots in order of priority influenced by whether there's a capable mentor or not for each topic.
6. Students must finally actually accept to do that project (some sign up for multiple organizations and then choose)
   - Sheesh!

If you are accepted as a mentor **and** you find a suitable student **and** we give you a slot **and** the student signs up for it, **then** you are an actual mentor.

It sounds harder than it is.

You get paid a modest stipend over the summer to mentor, typically $500 minus an org fee of 6%.

Several mentors donate their salary, earning ever better positions in heaven when that comes.

2017 Accepted Projects and Mentors

Alphabetic by project title

Student	Title	Mentor(s)
Karan Desai	A Model Zoo for Tiny-dnn	Edgar Riba, Stefano Fabri, Taiga Nomi
Laksono Kurnianggoro	API for Facial Landmark Detector	Antonella Cascitelli, Delia Passalacqua

Binbin Xu	Computational Occlusion Removal in Image Inpainting	Gary Bradski
Gang Song	Create Web-based Interactive Tutorials and Examples for OpenCV	Sajjad Taheri
João Cartucho	Documentation Improvement w/code examples	Vincent Rabaud
Suman Kumar Ghosh	End to End text detection and recognition	Prasanna
sukhad Anand	Face alignment with opencv	StevenPuttemans
Evgenii Zheltonozhskii	GPU enabled deep learning framework	Edgar Riba, Stefano Fabri
Mihai Bujanca	Implementing and extending DynamicFusion (Newcombe et al 2015)	Reza Amayeh, Zhe Zhang
Congxiang Pan	Improve and Extend the JavaScript Bindings for OpenCV	Sajjad Taheri
SHENGXIN QIAN	Improve Background Subtraction with Aggregated Saliency	Antonella Cascitelli
Vladislav Samsonov	Improvement of the background subtraction algorithm	Maksim Shabunin
kv	Learning compact models for object detection	Vladimir Tyan
Nan Yang	Photometric Calibration	Grace Vesom
Pau Rodríguez	Recurrent Neural Networks on tiny-dnn	Edgar Riba, Taiga Nomi
Jiri Horner	Speeding-up AKAZE features	Bence Magyar, Vadim Pisarevsky, Vladimir Tyan
Kuan Wang	The Fast Bilateral Solver	Bo Li

2017 Project Ideas:

Students may propose their own projects (give us a clear summary and why you can do this project). However, below are some of our priorities for this year Contact us and/or discuss ideas at https://groups.google.com/forum/?fromgroups#!forum/opencv-gsoc-2017

These are **not** in order of priority

1. Tutorials

We have so much good code in opencv_contrib that needs better documentation, tutorials, examples of use etc. Particularly:

- Our deep net library, tiny-dnn needs examples of how to
  - Setup, build and run on various OS
  - Training
  - Testing

    - Running on mobile
- DNN - examples of running models learned in Caffe and Torch. Speed tests
- Go over and update/improve/expand on existing tutorials
- Computational Photography
- Tracking
- Camera Calibration
- Video stabilization
- Image stitching

2a. Deep learning. Improving OpenCV Deep learning functionality

OpenCV works with/has an entire deep net library, tiny-dnn. Or, there is implemented from scratch DNN module. You are welcome to base your proposal on any of these 2 options. We want better performance, more tests, tutorials, python/java wrappers etc.

Improving DNN module ideas:

- Advanced visualizing of deep learning models (with input-output blobs dimensions, layer types, connections and so on). It can be either based on third-party libraries solution or built on OpenCV's drawing functions own implementation.
- Quantization of convolution and fully-connected layers. These layers are bottlenecks and by speeding them up it's possible to improve performance in many use cases.
- Enable supporting of the most popular deep learning architectures: VGG-16, ResNet, SqueezeNet, R-FCN and so on. Implement missed layers, check output accuracy, write samples for online-available models.
- Optimization of convolution layers - merging operations (Conv + bias + relu + pooing) to give more computation to one thread.

2b. Deep learning. Compact models.

We'd like to learn **compact** (that can be fit on mobile, i.e. models that take a few megabytes of disk or memory space) models (see SqueezeNet for inspiration that recognize:

- People
- Cars
- Animals, particularly dogs and cats outside and inside.
- Face and/or face features
- Text
- Floors, Walls, Ceilings, Windows, Doors, Tables, Chairs
- Others....

3. True, robust checkerboard out of April Tags or Aruco Tags

April Tags, Aruco Tags

The code for detecting Aruco and Charuco tags needs improvement.

4. Working, well-done SLAM

We have tried to get in reliable building blocks for SLAM for years. We've made some projects, but it's not there yet.

Please don't even think about doing this project unless you are already doing SLAM in your advanced PhD work. This is a hard algorithm to get right and you just won't unless you are already a leading expert!

5. Improve background subtraction

We have a Mixture of Guassian (MoG) fitting algorithms that work well, but they fail to take into consideration the correlation between neighbour pixels. Find a better way. We can suggest approaches.

Extra credit if it can detect or even compensate for camera motion.

6. Improve the video stabilization module

It works well now. Maybe just improve the tutorial, but try to improve the algorithm.

Try to add deep methods.

7. Saliency model

- Improve the saliency relevance
- Improve the application interface (API).

8. Text detection

- Improve the text detection module
- Test/improve on other datasets

9. Improve data labeling

Write better primitives for marking, segmenting, storing image (and possibly 3D!) data

But maybe just work with and improve the VATIC code (contact authors?)

- https://github.com/cvondrick/vatic
- http://web.mit.edu/vondrick/vatic/

10. OpenCV for interaction

- Hand or finger detection
- gesture recognition
- human skeleton based on 2d or 3d

11. Improve text recognition speed and accuracy in opencv_contrib/text module.

It would be interesting to adapt the module to use one of the several Deep CNN models have been made public during the last year (both for character and word recognition) e.g. http://www.robots.ox.ac.uk/~vgg/research/text/. Any other further improvement of the module is also welcome. Last year a DNN has been built that recognizes characters, but it's huge. We are interested in a very compact network (a few megabytes, may be up to 20Mb or so) for character recognition.

12. Extend translation of OpenCV into Java script. OpenCV.js

OpenCV.js OpenCV in javascript. Looks very interesting, has huge scale to run vision inside a browser

- https://github.com/ucisysarch
- Examples
  - http://ucisysarch.github.io/opencvjs/examples/img_proc.html
  - http://ucisysarch.github.io/opencvjs/examples/face_detect.html

13. Improve 3D visualization for vis

The vis module is very useful, but it needs:

- Easy way to directly enter cv::Mat containing depth or disparity into a point cloud for vis

- Point cloud to ply converter
- Once in vis, automatic color coded surface normal visualization see here
  - And also here
- Tutorials showing how to use its features
  - Add onto here
- Or, maybe we just need to document/improve and bring out functionality as in docs here
- Test codes for functionality are here

14. Optimizing various stuff

This is very general and always relevant topic. OpenCV is known for decent speed, but there is always room for improvement, especially in the experimental part (opencv_contrib). You are welcome to come up with concrete proposals on optimizing a particular functionality. Optimisation can be done using OpenCL, parallelisation, SSE/AVX or universal intrinsics. It can also be "algorithmic" optimization, which, despite its name, means that the algorithm basically stays the same, but you find some ways to eliminate some unnecessary computations, unnecessary memory allocations etc.

For example, we would be interested in optimizing image processing (imgproc, opencv_contrib/ximgproc), text detection (opencv_contrib/text), TLD tracker (opencv_contrib/tracking), computational photography (photo and opencv_contrib/xphoto), dnn module (opencv_contrib/dnn), feature detectors (features2d and opencv_contrib/xfeatures2d), non-dnn-based object detection (opencv_contrib/dpm and opencv_contrib/xobjdetect). You are welcome to propose for optimization some other functionality, though.

15. ROI algebra, improving mask creation and manipulation

Algebra on ROI-s implementing union, intersection, exclusion and more. Have to handle contours, rectangles and circles at least. Proposals are expected to expand on this abstract, come up with a fixed set of methods, be rich enough to justify for a full summer's work and demonstrate their usefulness with examples.

16. OpenCV in native Java Script: Opencv.js

A javascript version of OpenCV now has more than 1000 functions.

Yes! OpenCV henceforth will run natively in any browser

Help the authors expand, improve, document and create tutorials for it.

17. Halide language

- Optimizing exists algorithms or some parts using Halide.
- Extend implementations for GPU targets.
- Deep learning layers and topologies.
- Automatic domain-specific scheduling.

Potential mentors

- Alexander Mordvintsev zzznah(+)gmail.com
- Sid Bao ybao(+)magicleap.com
- Vincent Rabaud vincent.rabaud(+)gmail.com
- Vadim Pisarevsky vadim.pisarevsky(+)gmail.com
- Adrian Kaehler therealadrian(+)gmail.com
- Terry Boult tboult(+)vast.uccs.edu
- spmallick spmallick(+)taaz.com
- Serge Belongie sjb344(+)cornell.edu

- Stefano s.fabri10(+)gmail.com
- Prasanna pras.bits(+)gmail.com
- Pablo Alcantarilla pablofdezalc(+)gmail.com
- Bence Magyar mw.mzperx(+)gmail.com
- Manuele manuele.tamburrano(+)gmail.com
- Grace Vesom grace.vesom(+)gmail.com
- Open Source Computer Vision Library (OpenCV) garybradski(+)gmail.com
- Douglas Lee dougabug(+)gmail.com
- Claudia Rapuano c.rapuano(+)gmail.com
- Antonella Cascitelli antonellacascitelli(+)gmail.com
- Anatoly Baksheev anatoly.baksheev(+)itseez.com
- Alexander alexander.shishkov(+)itseez.com
- Alexander Smorkalov alexander.smorkalov(+)itseez.com
- Alexander Bovyrin alexander.bovyrin(+)itseez.com

In the below, get rid of the ''~~delete~~'' to make the emails work.

```
Anatoly Baksheev
Researcher, Vision Algorithms on GPU
Argus/Itseez
-delete-Anatoly.Baksheev@-delete-itseez.com

Alexander Bovyrin
PhD, Senior Researcher
Argus/Itseez founder
NNU Lecturer
-delete-alexander.bovyrin@-delete-itseez.com

Gary Bradski
Founder, "Arraiy":https://www.arraiy.com/
Founder, Industrial Perception Inc.
Former Consulting Prof. Stanford U.
OpenCV Founder, Technical Content Owner, GSoC Admin
Co-author of Learning OpenCV Book
https://www.amazon.com/Learning-OpenCV-Computer-Vision-Library/dp/1491937998
-delete-garybradski@-delete-gmail.com

Antonella Cascitelli
Grad student, University of Rome
-delete-antonellacascitelli@gmail.com

Eric Christiansen
PhD, UCSD
-delete-echristiansen@-delete-cs.ucsd.edu

Stefano Fabri
CRR Team leader, University of Rome
-delete-s.fabri10@gmail.com

Victor Eruhimov
OpenCV founding team/Senior Researcher
Argus/Itseez founder
```

NNU Lecturer
-delete-relrotciv@-delete-googlemail.com

Adrian Kaehler
Principle Engineer, Applied Minds
Co-author of Learning OpenCV Book.
-delete-therealadrian@-delete-gmail.com

Peter Karasev
PhD Student, MINERVA Research Group
Georgia Tech
-delete-karasevpa@-delete-gmail.com

Vadim Pisarevsky
OpenCV founding team/Czar
-delete-Vadim.-delete-Pisarevsky@-delete-gmail.com

Manuele Tamburrano
Grad Student, University of Rome
-delete-manuele.tamburrano@gmail.com

Vincent Rabaud
Senior Engineer, Google
-delete-vincent.rabaud@-delete-gmail.com

Claudia Rapuano
Grad student, University of Rome
-delete-c.rapuano@gmail.com

Grace Vesom
Senior Software Engineer, Magic Leap
-delete-grace.vesom@-delete_gmail.com

Bence Magyar
Pal Robotics
-delete-bence.magyar@-delete-pal-robotics.com

Pablo Alcantarilla
Toshiba Research Europe Ltd.
-delete-pablofdezalc@-delete-gmail.com

Back up Mentors

Mark Asbach
Fraunhofer IAIS
Schloss Birlinghoven
Sankt Augustin, Germany
http://mmprec.iais.fraunhofer.de/asbach.html
-delete-mark.asbach@-delete-iais.fraunhofer.de

Nicolas Saunier, Ph.D.

Assistant Professor
Civil, Geological and Mining Department (CGM)
École Polytechnique de Montréal
http://nicolas.saunier.confins.net
-delete-nicolas.saunier@-delete-polymtl.ca

Alexander Mordvintsev
Software Engineer
http://znah.net
-delete-zzznah@-delete-gmail.com

Andrey Morozov
Software Engineer
Argus/Itseez
-delete-andrey.morozov@-delete-itseez.com