

Bring FP16 type to OpenCV

- Author: Vadim Pisarevsky
- Links: [The feature request](#), [Another request](#)
- Status: **WIP** (<https://github.com/opencv/opencv/pull/10485>)
- Platforms: **All**
- Complexity: a few man-months (initial support can be added in less than 1 man-month)

Introduction and Rationale

16-bit floating-point is important data type in image processing and computer vision, especially in deep learning and in GPU-accelerated image processing and vision. We need to add such data type to OpenCV.

FP16 support in OpenCV should bring noticeable performance boost on the platforms that have hardware support for FP16, such as most of the modern GPUs.

In some special cases (like OpenCV DNN module) it's not absolutely necessary to have such support at the interface level. Instead, one can allocate arrays of `CV_16S` type (16-bit integers) and use them to store FP16 values. But it's a hack, and in many other cases, e.g. image processing pipelines, users should be able to explicitly specify the actual type of their data and process it accordingly.

Proposed solution

- Add `CV_16F` type with value 7 (currently used for `CV_USRTYPE1`, which does not have a fixed interpretation and is not supported by any array processing function).
- Extend basic matrix allocation and manipulation function to support this type. Arithmetic operations should do temporary conversion to FP32 and back, if needed. Some may report `Error_StsNotImplemented` error, which is what they already do by default.
- Extend XML/YAML persistence and other utility functionality to support this type.
- Extend some dynamically constructed OpenCL kernels to support this new type. Q: what to do if GPU does not support FP16?
- Extend DNN to include FP16 processing path (see the dedicated [Evolution Proposal](#))

Impact on existing code, compatibility

Currently, `CV_USRTYPE1` is practically out of use, so all the existing code should still work.

Possible alternatives

Temporarily use `CV_16S` / `CV_16U` as placeholder for FP16 data. This is the currently used solution.

References

TBD