پردازش تصویر و واقعیت افزوده با opencv در C#

نویسنده :

محمد نیک‌روش

E-mail: *Hootan09@gmail.com*

با تشکر از سایت

Ehsanavr.com

جهت جلوگیری از کپی ، تمامی صفحات watermark شده است.

Ehsanavr.com

شهریورماه ۱۳۹۳

Ehsanavr.com

Ehs

تقدیم به:

آنان که در راه کسب دانش راهنمایم بودند .

Ehsanavr.com

تشکر و قدردانی:

با تقدیر و درود فراوان خدمت پدر و مادر بسیار عزیز ، دلسوز و فداکارم که پیوسته جرعه نوش جام تعیلم و تربیت ، فضیلت و انسانیت آنها بوده ام و همواره چراغ وجودشان روشنگر راه من در سختی ها و مشکلات بوده است.

و

با تشکر و سپاس از استاد دانشمند و پر مایه ام جناب آقای مهندس علیرضا نعیمی صدیق که از محضر پر فیض تدریسشان ، بهره ها برده ام.

Ehsanavr.com

چکیده

پردازش تصویر یکی از علوم پرکاربرد در مباحث مربوط به علوم کامپیوتر است .در این پروژه سعی شده است از راحت ترین و در دسترس ترین کتابخانه های موجود در پردازش تصویر همچون opencv و زبان های برنامه نویسی چون #c جهت سهولت در برنامه نویسی و تمرکز فرد بر روی روش پیاده سازی پروژه استفاده شود.

امروزه این علم در صنعت و صنایع دفاعی به صورت عمده استفاده می شود و مثال واقعی از این علم در تجارت ، مربوط به دوربین های عکاسی می شود که به صورت خودکار چهره فرد را تشخیص داده و کادری دور صورت فرد ترسیم می کند.

استفاده این علم به جهت ایجاد واقعیت افزوده در برنامه ها و بازی های کامپیوتری در سال های اخیر بیشتر دیده می شود که ما در فصل انتهایی اشاره ای کوتاه به آن خواهیم نمود.

در این پروژه سعی شده که تمامی مثال ها در قالب Windows Form Applicationباشد تا افراد تمرکز خود را بر روی درک مفاهیم پردازش تصویر معطوف کنند.

فصل های مربوط به این پروژه مکمل و تکمیل کننده فصول بعد می باشد ، به همین جهت از خوانندگان گرامی تقاضا دارم که فصل هارا به ترتیب و به صورت متوالی دنبال کنند تا در درک مفاهیم و موضوعات در هر یک از فصل ها دچار اشکال نشوند.

علت انتخاب این موضوع تحقیق این بود که بررسی کنیم که صنعت کامپیوتر چه دستاورد هایی در برنامه نویسی پردازش تصویر و واقعیت افزوده در سال های اخیر داشته است و باتوجه به افزیش سرعت کامپیوتر های امروزی آیا قادر خواهیم بود تا به کامپیوترها بیاموزیم که مانند انسان ها ببینند .

از طرفی دیگر این تحقیق در رابطه با اضافه کردن اشیا (در این پروژه ، تصاویر) به محیط واقعی ، مطالعات و مثال هایی را ارائه می دهد.

واژه‌های کلیدی:پردازش تصویر – واقعیت افزوده – کامپیوتر

Ehsanavr.com

فهرست مطالب

فصل ۱: مقدمه **۱**

۱-۱- مقدمه ۲

فصل ۲: مروری بر منابع **۴**

۲-۱- مقدمه ۵

۲-۲- مروری بر ادبیات موضوع ۵

فصل ۳: opencv چیست؟ **۸**

۳-۱- مقدمه ۹

۳-۲- اجزا و کلاس های opencv ۱۱

۳-۳- پلتفرم ها و سخت افزار های قابل پشتیبانی ۱۲

۳-۴- نتیجه گیری ۱۴

فصل ۴: نصب و پیاده سازی opencv در visual studio 2010 **۱۵**

۴-۱- مقدمه ۱۶

۴-۲- نحوه نصب (در قالب تصویر) ۱۶

۴-۳- اتصال opencv به visual studio ۱۹

۴-۴- نتیجه گیری ۲۴

فصل ۵: انواع و متغیرهای اصلی در opencv **۲۵**

۵-۱- مقدمه ۲۶

۵-۲- انواع و متغیرهای اصلی ۲۶

۵-۲-۱- انواع داده در تصویر ۲۸

۵-۲-۲- دسترسی به عناصر تصویر با cvGet2D و cvSet2D ۲۸

۵-۲-۳- توضیحات مربوط به کد اولیه برنامه (نمایش یک تصویر) ۲۹

۵-۳- بررسی کتابخانه opencv در دیگر زبان های برنامه نویسی و سطح بالا ۳۱

۵-۴- نتیجه گیری ۳۳

فصل ۶: استفاده از Opencv در یک Windows Form Application در ویژوال استدیو به زبان C# **۳۴**

۶-۱- مقدمه ۳۵

۶-۲- ایجاد پروژه و طراحی فرم ۳۵

۶-۳- اضافه کردن dll های مربوطه (opencvsharp) ۳۷

۶-۴- شروع برنامه نویسی ۳۸

۶-۵- نتیجه گیری ۴۱

Ehsanavr.com

فصل ۷: تطبیق الگو با استفاده از opencv در Windows Form Application C# ۴۲
۷-۱- مقدمه ۴۳
۷-۲- توضیحات مربوط به کد و برنامه نویسی ۴۴
۷-۳- نتیجه گیری ۵۰

فصل ۸: واقعیت افزوده با opencv در Windows Form Application C# ۵۱
۸-۱- مقدمه ۵۲
۸-۲- واقعیت افزوده چیست؟ ۵۲
۸-۲-۲- کاربرد ها ۵۳
۸-۳- ایجاد یک نمونه ساده از پیاده سازی واقعیت افزوده ۵۴
۸-۴- نتیجه گیری ۶۳

فصل ۹: جمع‌بندی و پیشنهادها ۶۴
۹-۱- مقدمه ۶۵
۹-۲- محتوا ۶۵
۹-۲-۱- جمع‌بندی ۶۵
۹-۲-۲- نوآوری ۶۶
۹-۲-۳- پیشنهادها ۶۶

مراجع ۶۷

پیوست‌ها ۶۹

Ehsanavr.com

فهرست اشکال

شکل (۲-۱) تصویر لوکاس ۷
شکل (۳-۱) توضیحات مربوط به opencv ۹
شکل (۳-۲) روند رشد تاریخی این کتابخانه ۱۰
شکل (۳-۳) روابط بین کتابخانه های opencv ۱۲
شکل (۳-۴) پلتفرم های قابل پشتیبانی opencv ۱۳
شکل (۴-۱) دانلود کتابخانه ۱۶
شکل (۴-۲) مراحل نصب ۱۷
شکل (۴-۳) مراحل نصب ۱۷
شکل (۴-۴) مراحل نصب ۱۷
شکل (۴-۵) مراحل نصب ۱۸
شکل (۴-۶) مراحل نصب ۱۸
شکل (۴-۷) مراحل نصب ۱۸
شکل (۴-۸) مراحل نصب ۱۹
شکل (۴-۹) مراحل نصب ۲۰
شکل (۴-۱۰) مراحل نصب ۲۰
شکل (۴-۱۱) مراحل نصب ۲۱
شکل (۴-۱۲)مراحل نصب ۲۱
شکل (۴-۱۳) مراحل نصب ۲۲
شکل (۴-۱۴) مراحل نصب ۲۳
شکل (۴-۱۵)خروجی برنامه به زبان C++ ۲۴
شکل (۵-۱) مثال کاربردی ۲۹
شکل (۵-۲) کتابخانه EmguCV ۳۲
شکل (۶-۱) نمایش تصویر در Windows form ۳۵
شکل (۶-۲) طراحی فرم ۳۶
شکل (۶-۳) محیط کد زنی ۳۶
شکل (۶-۴) اضافه کردن Dll های مربوطه ۳۷
شکل (۶-۵) دکمه Stop در فرم ۴۰
شکل (۶-۶) ذخیره سازی تصویر ۴۱
شکل (۷-۱) محدودیت سرعت ۵۵ کیلومتر ۴۳
شکل (۷-۲) تطبیق الگو ۴۳
شکل (۷-۳) تابلو ۴۴

Ehsanvr.com

شکل (۷-۴) خروجی تطبیق الگو ۴۸
شکل (۸-۱) واقیت افزوده ۵۳
شکل (۸-۲) ایجاد پروژه در ویژوال استودیو ۵۴
شکل (۸-۳) الگوی معرفی شده به برنامه ۵۷
شکل (۸-۴) تصویر نویسنده ۵۸
شکل (۸-۵) خروجی برنامه ۵۸
شکل (۸-۶) خروجی نهایی برنامه ۶۰
شکل (۸-۷) خروجی نهایی برنامه ۶۳

Ehsanavr.com

فهرست جداول

جدول (۵-۱) انواع داده در تصویر ۲۸

Ehsanavr.com

فهرست علائم اختصاری

Ehsanavr.com

فـصل ۱ :

مقدمه

Ehsanavr.com

۱-۱- مقدمه

جامعه صنعتی قرن بیستم امروزه جای خود را به جامعه اطلاعاتی قرن بیست و یکم داده است به همین دلیل بکارگیری فناوری های نوین از جمله فناوری اطلاعات و ارتباطات وعلوم کامپیوتر در امور مختلف زندگی امری اجتناب ناپذیر است که باید به طور هدفمند بکار گرفته شود.
پوشیده نیست که دگرگونی بنیادی و پیشرفت چشمگیر در عالی ترین سطوح هر جامعه با هر گرایش ملی ، سیاسی ، اعتقادی و ... ، نتیجه مستقیم تحول ریشه ای و پیشرفت تکنولوژی است .

باتوجه به پیشرفت هایی که در سال های اخیر مربوط به علوم کامپیوتر[۱] و صنایع الکترونیک[۲] به وجود آمده است ، امروزه به شدت نیاز است که ابزار های کاربردی بشر پیچیده تر و محدوده استفاده افراد از ابزار های وابسته به کامپیوتر بیشتر شود .یکی از این مباحث مورد نیاز پردازش تصویر و واقعیت افزوده است که سعی بر آن دارد که کیفیت زندگی افراد را به سطوح بالاتری ارتقا دهد.
نمونه بارز آن عینک های گوگل است که گام بلندی در زمینه پردازش تصویر و علوم بینایی سنجی ماشین برداشته است.
من به شخصه وقتی برای اولین بار با پردازش تصویر و برنامه نویسی پردازش تصویر آشنا شدم ،دیدن برنامه ای مربوط به نرم افزار متلب بود که در آن شخصی با تکان دادن خودکار خود در مقابل دوربین ، برنامه را مجاب به دنبال کردن مسیر حرکت خودکار فرد می کرد.
این برنامه بسیار جالب بود ولی وقتی که به شخصه اقدام به برنامه نویسی با متلب کردم متوجه کندی زیاد این محیط برنامه نویسی شدم.
از طرفی تحریم های نرم افزاری و از همه مهم تر تحریم سایت اصلی آن که مثال های کاربردی و سوالات اشخاص را پاسخ می داد ، برای کاربران ایرانی فیلتر شده بود و من از این موضوع بسیار آشفته و ناراحت شدم ، پس شروع به تحقیق در اینترنت کردم تا دریابم که چه کتابخانه ها و محیط های نرم افزاری دیگری در زمینه پردازش تصویر موجود است که مشکلات نرم افزار متلب را نداشته باشد .
در این میان بود که با کتابخانه رایگان و منبع باز Opencv آشنا شدم این کتابخانه هر چند مدت یکبار

---

1 Computer science

i2 electronics

Ehsanavr.com

توسط گروه های کاربری توسعه داده شده و بروز رسانی می شود.

از همه مهم تر سرعت بالا و حجم کم این کتابخانه نظر این حقیر را جلب کرده است .

من وقتی بیشتر به آن علاقمند شدم که فهمیدم گروه تحقیق و توسعه اینتل بر روی آن کار می کند و اولین بار افرادی از این شرکت بودند که این کتابخانه را توسعه دادند.

نسخه های اول این کتابخانه به طور کامل به زبان C نوشته شده بود و در نسخه های بعدی با زبان C++ نیز پورت شد و توسعه آن در دیگر زبان های برنامه نویسی نیز میسر شد.

نکته جالب دیگر این برنامه این بود که سعی شده بود تمامی مراحل برنامه نویسی را به صورت مجزا در داخل کتابخانه های خود جای دهد و حتی به کاربران اجازه استفاده از توابع ورودی و خروجی جهت خواندن و نوشتن در حافظه را به صورت مستقیم و بدون اضافه کردن کتابخانه های دیگر در برنامه را می داد. اما مشکلاتی هم در این کتابخانه وجود داشت که یکی از عمده ترین آنها این بود که این کتابخانه تعاریف و انواع مختص به خود را داشت لذا تبدیل این انواع به انواع دیگر کاری بس دشوار و مشکل بود و حافظه مصرفی کامپیوتر برای این تبدیل انواع رو به ازدیاد می گذاشت.

یکی دیگر از این مشکلات ، نبود کد های سطح بالا بود که خوشبختانه ، گروهی اقدام به تبدیل این کتابخانه از زبان C++ به زبان C# کردند ، که این کار کیفیت برنامه نویسی و سهولت نوشتن برنامه را بسیار بالا می برد.

این پروژه (پردازش تصویر در زبان های سطح بالا همچون سی شارپ) ابتدا در پروژه گوگل کد مطرح شد و بعد ها و پس از به نتیجه مطلوب رسیدن آن ، کل پروژه به سایت Github منتقل شد تا افراد بتوانند تمامی آن را به صورت یکجا بر روی کامپیوتر خود دانلود کرده و از آن استفاده نمایند.

از کاربرد های پردازش تصویر می توان به صنایع فیلم اشاره نمود و به نرم افزار هایی همچون After effect و فتوشاپ اشاره کرد که زمینه هایی را برای کسب درآمد از کار بر روی تصاویر و ساخت کلیپ و فیلم ایجاد نموده است .

نکته جالب تر آنکه حتی فرمت های ویدیویی و تصویری همچون jpg و یا Avi در درون ساختار خود از پردازش تصویر استفاده می کنند .

مثلا فرمتی مثل Avi در درون خود پارامتر هایی همچون pixel rate و frame rate دارد که این پارامتر ها کیفیت و سرعت تصاویر و ... را در درون این فرمت ها مشخص می کنند.

در صنعت و صنایع دفاعی و نظامی نیز به دلیل آنکه بحث سرعت در پردازش تصویر مطرح است ، برنامه نویسان اغلب به برنامه نویسی با زبان های سطح پایین تر همچون C و C++ اقدام می کنند وافراد در برنامه نویسی تجاری و کاربردی از C# و جاوا استفاده می کنند و این آزادی عمل امکان توسعه در این علم را به بهترین روش ممکن فراهم می سازد.

Ehsanavr.com

فـصل ۲ :

مروری بر منابع

Ehsanavr.com

۲ – ۱ – مقدمه

متاسفانه به علت اینکه تحقیق انتخاب شده موضوع جدیدی (برنامه نویسی opencv در C#) را مطرح می کند و از زمان انتشارکتابخانه های آن (opencvsharp) مدت کمی می گذرد لذا منابع چندانی در دسترس نیست وتوضیحات مربوط به این فصل بسیار محدود و مختصر خواهد بود.

همان طور که در مقدمه ذکر شد ، کل پروژه به سایت Github منتقل شده است و این کار توسط یک کاربر از کشور ژاپن انجام شده است. این کاربر Shimat نام دارد . وی بر روی زمینه هایی از کار با پردازش تصاویر کار می کند.

همه پروژه های این شخص مد نظر ما نیست و بلکه تنها در زمینه opencvsharp از آن بهره گرفته شده است .

تا به امید خدا ما نیز بتوانیم قدمی هر چند ناچیز در این زمینه و ارتقا این علم نو برداریم.

۲ – ۲ – مروری بر ادبیات موضوع

از ویژگی هایی که این فرد (Shimat) در کتابخانه Opencvsharp به آنها اشاره نموده است ، چند سکویی بودن این کتابخانه است. از دیگر مشخصه های بارز این کتابخانه میتوان به موارد زیر اشاره کرد.

Features

- OpenCvSharp is modeled on the native OpenCV C/C++ API style as much as possible.
- Many classes of OpenCvSharp implement IDisposable. There is no need to manage unsafe resources.
- OpenCvSharp does not force object-oriented programming style on you. You can also call native-style OpenCV functions.
- OpenCvSharp provides functions for converting from Mat/IplImage into Bitmap (GDI+) or WriteableBitmap (WPF).
- OpenCvSharp can work on **Mono**. It can run on any platform which **Mono** supports (e.g. Linux and MacOSX).

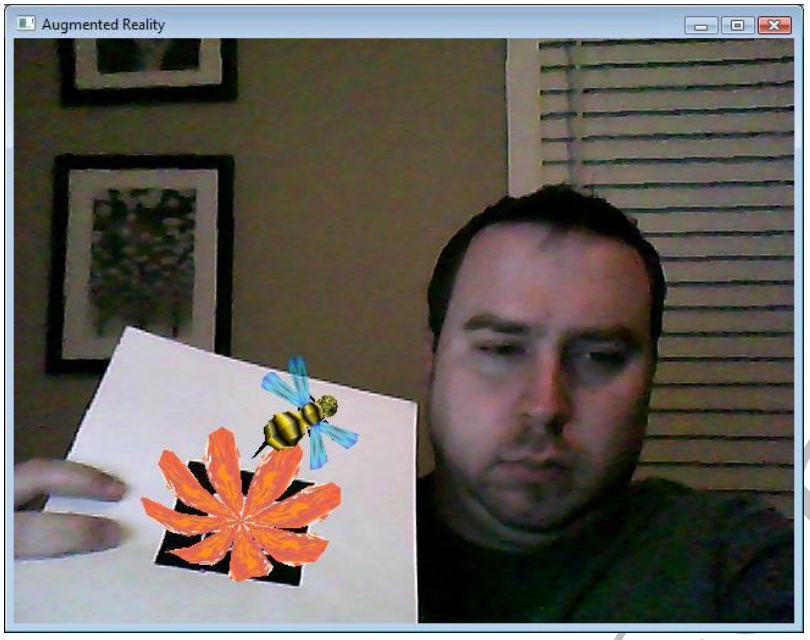
Ehsanavr.com

تنها یکی دیگر از افراد ، زمینه تحقیقاتی خود را در رابطه با این کتابخانه انجام داده است که ایشان شخصی از کشور انگلستان می باشند که در ابتدا روند تحقیقاتی خود را در زمینه پردازش تصویر با زبان برنامه نویسی پایتون آغاز کرده اند و در انتها مقالات اندکی و به طور کاملا خلاصه در زمینه پردازش تصویر با C# نوشته اند و این مقالات بسیار کاربردی بوده و مقدار زیادی از ارجاعات این پروژه مربوط به همین مقالات می باشد . ولی متاسفانه این مقلات به جنبه های کاربردی قضیه پرداخته است و توضیحات علمی آن بسیار محدود است. این مقالات به شرح زیر می باشند.

- OpenCV Edge Detection and Video Capture with C#
- OpenCV Head Tracking with C#
- Record Video with OpenCV and C#
- OpenCV Corner Detection with C#
- OpenCV Eye Tracking with C#
- OpenCV Skin Detector with C#
- Use OpenCV in a Windows Form Application in C#
- More Eye Tracking with OpenCV and C#
- Template Matching with OpenCV and C#
- Augmented Reality Video
- How to Train OpenCV Haar Classifiers
- Augmented Reality Using C# and OpenCV
- Getting Started with OpenCV and Python
- Introduction to embedded vision and the OpenCV library
- Virtual 3D with OpenCV and C#
- Lane Detection with OpenCV and C#

آدرس این سایت **http://www.learncomputervision.com** می باشد ولی به مدت زیادی این سایت به روز رسانی نشده است. نام این فرد (صاحب سایت) لوکاس است و آخرین اخبار از این شخص این است که در زمینه اپلیکیشن نویسی برای پلتفرم های همراه فعالیت دارد .

Ehsanavr.com



شکل (۲-۱) تصویر لوکاس

Ehsanavr.com

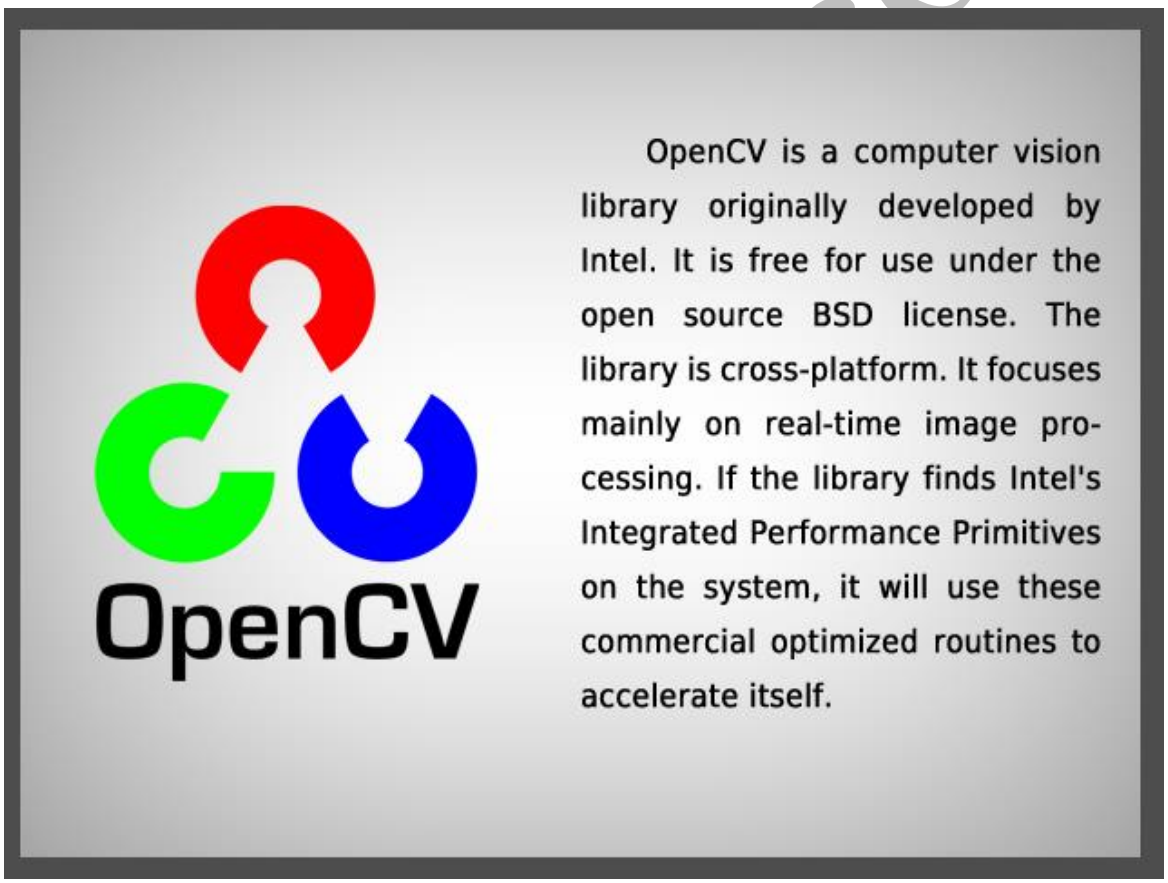
فـصل ۳ :

opencv چـیـست؟

Ehsanavr.com

۳-۱- مقدمه

Opencv یک کتابخانه پردازش تصویر حرفه ای است که نسخه رسمی آن برای اولین بار توسط تیم تحقیقاتی اینتل و در آزمایشگاه هوش مصنوعی این شرکت تولید شد .این آزمایشگاه یکی از شعبات شرکت اینتل در کشور روسیه است که توسط شخصی به نام وادیم پیزاروسکی مدیریت می شود.
مکان شروع برنامه نویسی این کتابخانه اولین بار از دانشگاه ام ای تی شروع شد و دست به دست بین دانشجویان می چرخید تا اینکه بدست گروه تحقیق و توسعه اینتل رسید و این شرکت وظیفه توسعه و بهینه سازی آن را به آزمایشگاه روسی خود محول کرد.



شکل (۳-۱) توضیحات مربوط به opencv

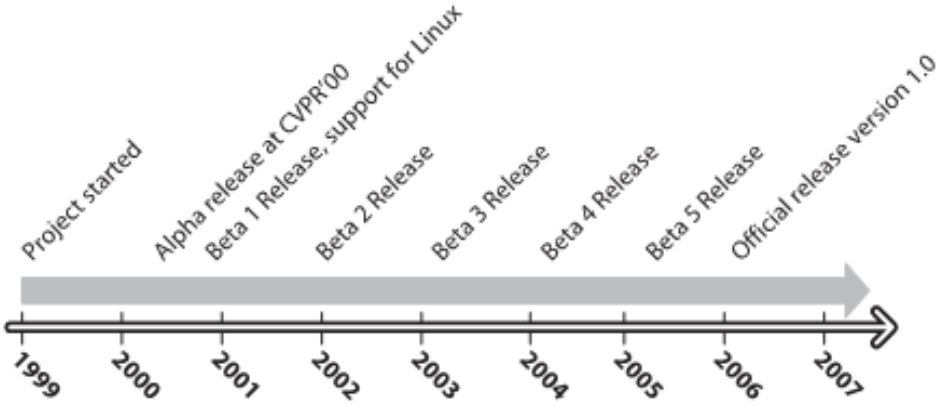
Opencv به معنی open source computer vision می باشد و کتابخانه های متفاوتی جهت پردازش حرفه ای ویدیو و تصاویر را در بر دارد.

این کتابخانه دارای مجموعه ای از روال ها و ابزارها است که می تواند قابلیت هایی نظیر:

- Facial recognition system
- Gesture recognition
- Human-computer interface
- Mobile robotics
- Segmentation and recognition
- Motion tracking
- Artificial neural networks
- ....

برای برنامه نویسان ایجاد کند.

نسخه اولیه این کتابخانه در سال ۱۹۹۹ معرفی شد و هنوز هم به طور رایگان توسعه داده می شود.



شکل (۳-۲) روند رشد تاریخی این کتابخانه

زبان هدف در opencv ، زبان C بوده است اما از نسخه ۲,۰ به بعد به صورت مستقیم C++ را هم پشتیبانی می کند.

هم اکنون آخرین نسخه آن (opencv 2.4.9) است و به صورت رایگان در آدرس :

http://opencvlibrary.sourceforge.net

موجود و شامل توضیحات کامل و مثال های عملی می باشد.

Ensaf.com

۳-۲- اجزا و کلاس های opencv

Opencv از ۵ کتابخانه اصلی تشکیل شده است که در کنار هم مجموعه ی کلی کتابخانه پردازش تصویر را تشکیل می دهند و شامل موارد زیر است :

- cvLibrary
- aux Library
- cxCoreLibrary
- highgui Library
- Machine learning Library(MLL)

۱. cvLibrary

شامل کتابخانه های پردازش تصویر و ویدیو و الگوریتم های آن می باشد.

۲. auxLibrary

برای سطوح بالای پردازش تصویر مورد استفاده قرار می گیرد و مباحثی شامل :

- بینایی استریو
- ردیابی دهان وچشم
- ردیابی سه بعدی
- جداسازی پیش زمینه – پس زمینه
- کالیبراسیون دوربین
- و...

را پوشش می دهد ، اما متوسفانه منابع زیادی در رابطه با آن موجود نیست.

۳. cxCoreLibrary

ساختارها و الگوریتم های پایه از پردازش تصویر را دارا بوده و رابط بین کد های شما با سیستم عامل و کارت گرافیک سیستم می باشد.

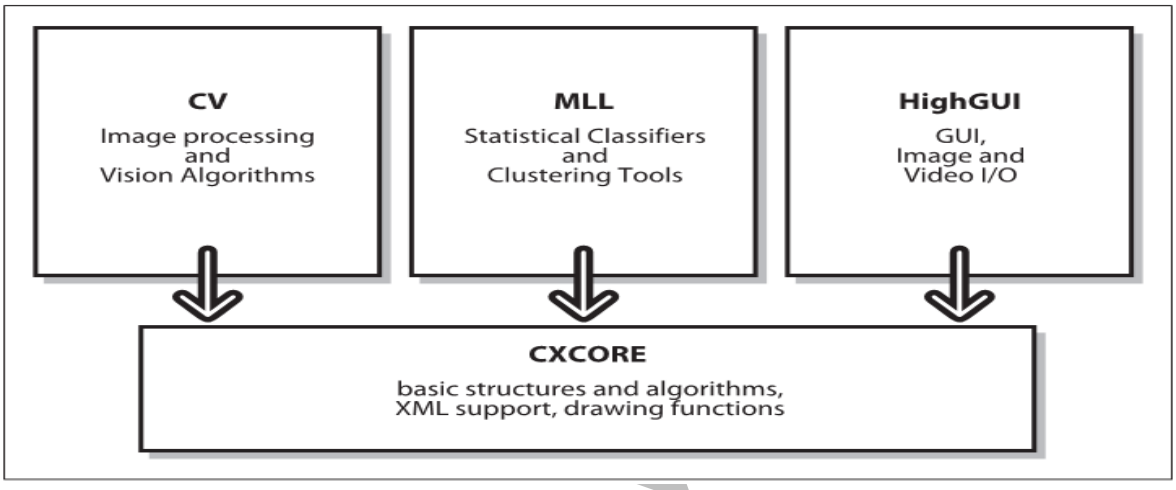
Ehsanavr.com

۴. highGuiLibrary

شامل روتین ها و توابع IO برای ذخیره سازی و بارگذاری تصاویر و ویدیو است.

۵. (MLL) Machine learning Library

شامل کتابخانه یادگیری ماشین و کلاس بندی های آماری و ابزار خوشه بندی است.



شکل (۳-۳) روابط بین کتابخانه های opencv

۳-۳- پلتفرم ها و سخت افزار های قابل پشتیبانی

این کتابخانه پردازش تصویر از آنجایی که با زبان های C و C++ نوشته و توسعه داده شده است ، قابلیت اجرا روی تمامی سیستم عامل ها و سخت افزار ها از جمله:

- Linux
- Windows
- Mac
- Iphone
- Android
- و...

را داراست که در ادامه سعی خواهد شد برنا.مه هایی را بر روی پلتفرم ویندوز و در قالب مثالی کاربردی پوشش دهیم.

Ehsana

به عکس زیر دقت کنید:
این عکس توضیحات کاملی در رابطه با پشتیبانی opencv از انواع معماری ها بیان می کند.
ستون های IA32 معماری ۳۲ بیتی معمول و IA64 cpu های ۶۴ بیتی معمول را نشان می دهد.

	IA32	EM64T	IA64	Other (PPC, Sparc)
Windows	✓ (w. IPP; MSVC6, .NET2005+OMP, ICC, GCC, BCC)	✓ (w. IPP; MSVC6+PSDK.NET2005+OMP, PSDK)	± (w. IPP; PSDK, some tests fail)	N/A
Linux	✓ (w. IPP; GCC, BCC)	✓ (w. IPP; GCC, BCC)	✓ (GCC, ICC)	✗
MacOSX	✓ (w. IPP, GCC, native APIs)	? (not tested)	N/A	✓ (iMac G5, GCC, native APIs)
Others (BSD, Solaris…)	✗	✗	✗	Reported to build on UltraSparc Solaris

شکل (۳-۴) پلتفرم های قابل پشتیبانی opencv

ممکن است در عکس بالا کلمه IPP توجه شما را جلب کند.IPP در حقیقت یک کتابخانه پولی است که در زمینه پردازش تصویر به opencv الحاق می شود.
شنیده ها حاکی از ان است که IPP تماما با زبان اسمبلی نوشته و توسعه داده شده است و بنابراین از سرعت بسیار بالایی برخوردار است و حتی شرکت اینتل برای این کتابخانه (IPP) ، سخت افزار خاصی تولید کرده که برای مقاصد حرفه ای قابل تهیه از بازار است .
اما این کتابخانه روی تمامی cpu های خانواده x86 هم به خوبی کار می کند.
در آزمایشی که توسط یکی از شرکت ها انجام شد ، معلوم شد که IPP روی cpu های AMD حتی بسیار بهتر از اینتل عمل می کند، که این شگفتی بسیاری از برنامه نویسان را برانگیخت.

عکس بالا توضیحات کامل تری را نیز می طلبد که خارج از مباحث این مقاله می باشد .
برای توضیحات بیشتر می توانید به کتاب Learning Opencv نوشته Gray bradski & Adrian Kaehler مراجعه کنید که در ضمیمه این مقاله موجود است .البته نسخه ترجمه شده این کتاب نیز وجود دارد که به راستی ارزش خریدن هم ندارد. چون این کتاب هم بد نوشته شده است و هم مترجم آن را بد ترجمه کرده است.(به عنوان کسی که این کتاب را خریده و خوانده است توصیه نمی کنم).

Ehsana

بهترین منابع در خود سایت opencv موجود است و احتیاج به خواندن هیچ منابع اضافی برای **برنامه نویسی** را ندارد.

Ehsanavr.com

۳-۴- نتیجه گیری

استفاده از این کتابخانه روند برنامه نویسی در پردازش تصویر را بسیار تسهیل می کند زیرا این کتابخانه تمامی ابزارهای لازم برای برنامه نویسی را دارد و دیگر نیاز به اضافه کردن هیچ کتابخانه خارجی دیگری نیست. هرکدام از اجزای این کتابخانه در قالب یک کتابخانه داینامیک (dll) تمامی نیازهای برنامه نویس را رفع می کند .

فـصل ۴ :

نصب و پیاده سازی opencv در visual studio 2010

Ehsanavr.com

۴-۱- مقدمه

جهت نصب opencv برای برنامه نویسی به زبان C و یا C++ نیاز است که یکسری از تغییرات لازم را در سیستم عامل و محیط برنامه نویسی (IDE) ایجاد کرد تا بتوان از این مجموعه کتابخانه استفاده های لازم و کافی را برد . نکته مهم این است که این کتابخانه را می توان بر روی تمامی پلتفرم ها و محیط های برنامه نویسی بکار برد ، در صورت نیاز می توان با کمک و راهنمایی سایت مربوطه ، اقدام به نصب و راه اندازی آن در دیگر محیط های برنامه نویسی کرد.

http://opencv.org

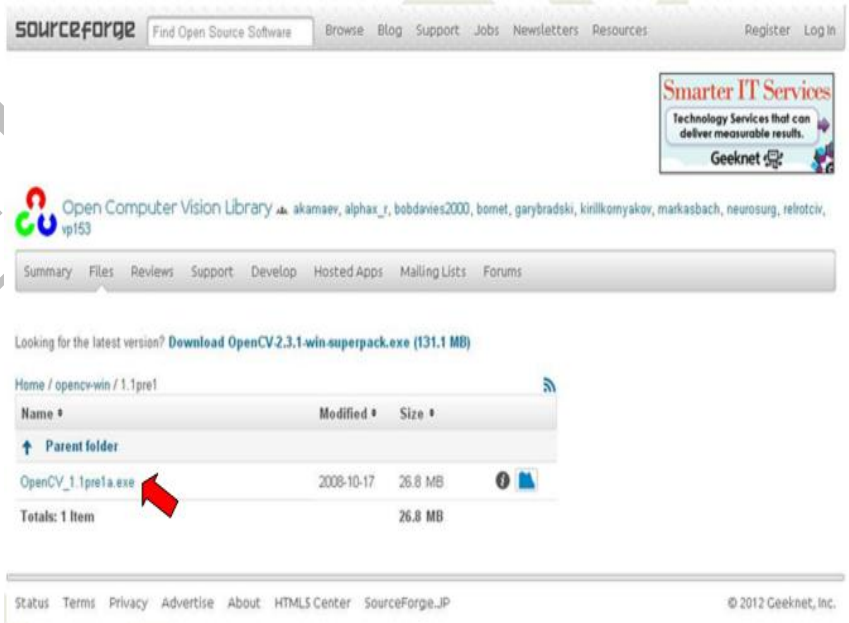
توجه:

در این مقاله تمامی پروژه ها در visual studio و با نسخه های ۲۰۱۰ و ۲۰۱۳ نوشته و اجرا می شوند . ولی روش کار روی تمامی نسخه ها به همین منوال است .

در صورت نیاز به توضیحات اضافی ، آموزش هایی در قالب ویدیو در فایل ضمیمه ، ارائه خواهد شد.

۴-۲- نحوه نصب (در قالب تصویر)

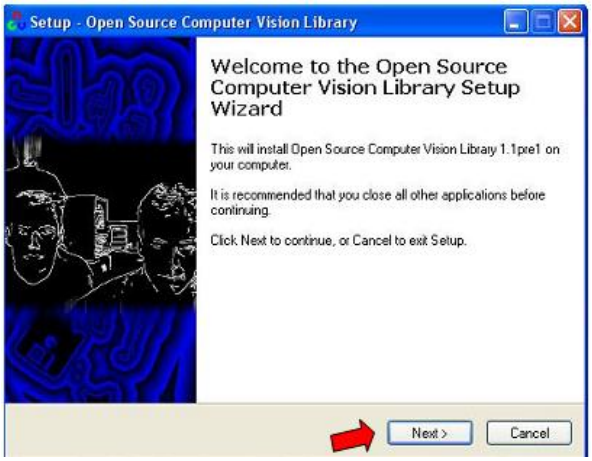
پس از دانلود کتابخانه از سایت مربوطه داریم :



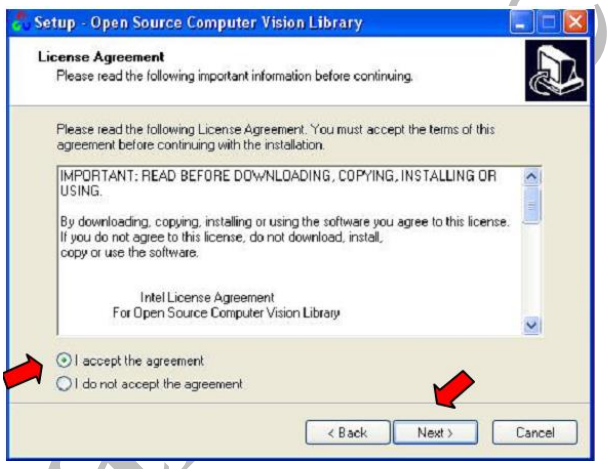
شکل (۴-۱) دانلود کتابخانه

F'anavr.com

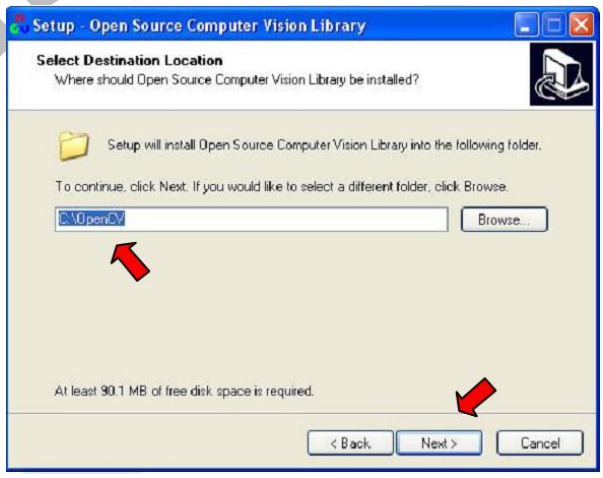
فایل exe را اجرا کرده و مراحل را طبق تصاویر ادامه می دهیم.



شکل (۴-۲) مراحل نصب



شکل (۴-۳) مراحل نصب



شکل (۴-۴) مراحل نصب

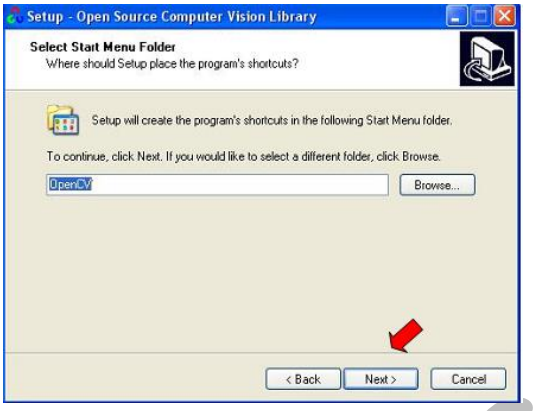
Ehsa...com

نکته :

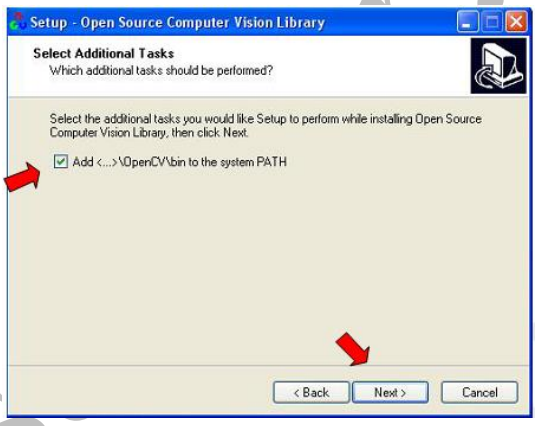
این کتابخانه ، هم در قالب فایل نصبی و هم به صورت فایل آرشیو ، قابل دانلود و استفاده است .

توجه کنید که برای سهولت نصب و اجرا مسیر اجرایی را به صورت :

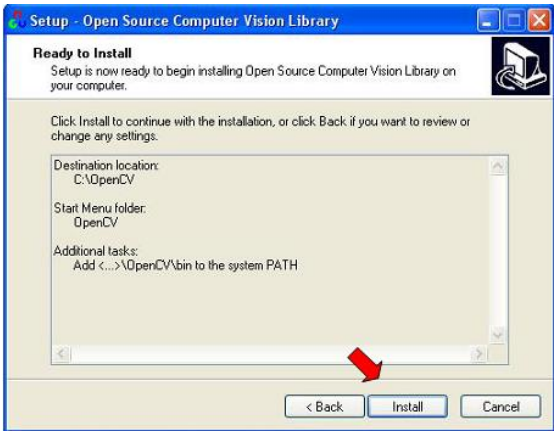
C:\OpenCv قرار دهید.



شکل (۴-۵) مراحل نصب



شکل (۴-۶) مراحل نصب



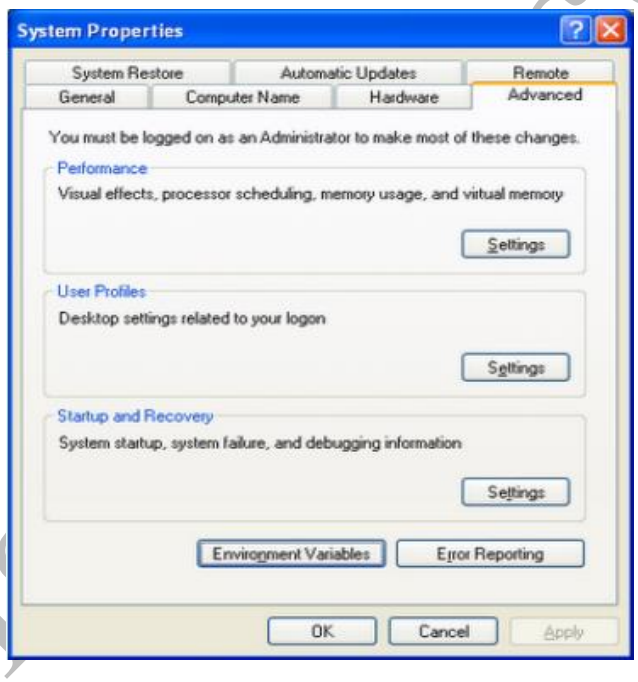
شکل (۴-۷) مراحل نصب

Ehsa...com

وقتی مراحل نصب تکمیل شد اینک نیاز است که opencv را به ویژوال استدیو متصل کنیم تا بتوانیم برنامه نویسی را در محیط ویژوال استدیو را شروع کنیم.

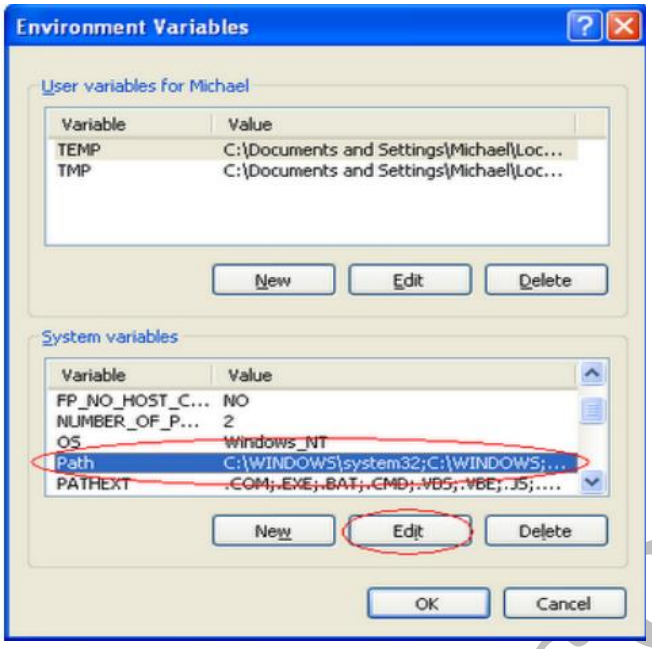
۴-۳- اتصال opencv به visual studio

روی my computer راست کلیلک کنید و properties را بزنید و از سمت چپ گزینه Advance System Setting را انتخاب کنید.
سپس در پنجره باز شده تب Advance را انتخاب و به قسمت "Environment Variables" بروید.



شکل (۴-۸) مراحل نصب

Ebook.com



شکل (۴-۹) مراحل نصب

اینک از قسمت system variables گزینه path را انتخاب و Edit را بزنید.

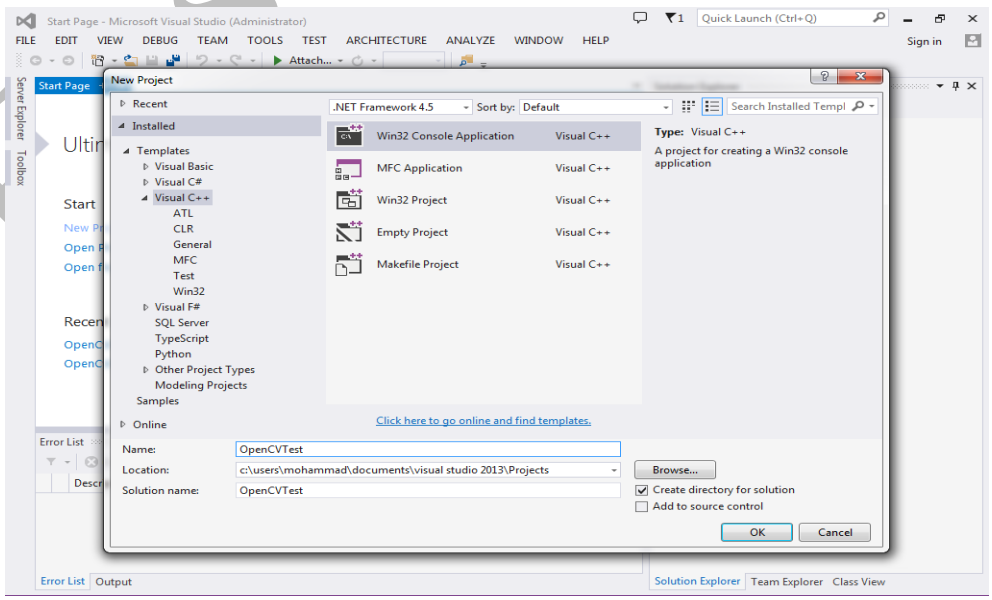
اکنون برای سیستم های ۳۲ بیتی مسیر زیر را به path اضافه کنید:

c: \opencv\build\x86\vc10\bin

درصورت استفاده از سیستم های ۶۴ بیتی مسیر زیر را به path اضافه کنید:

c: \opencv\build\x64\vc10\bin

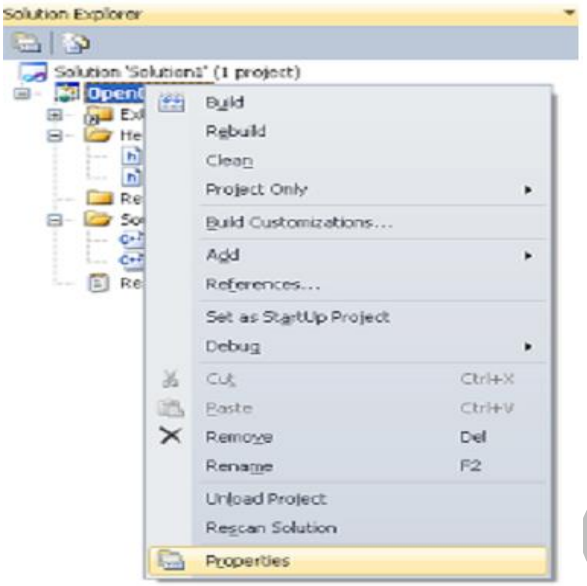
سپس در ویژوال استودیو یک پروژه c++ از نوع Win32 console application بسازید و نامگذاری کنید برای مثال: OpenCVTest



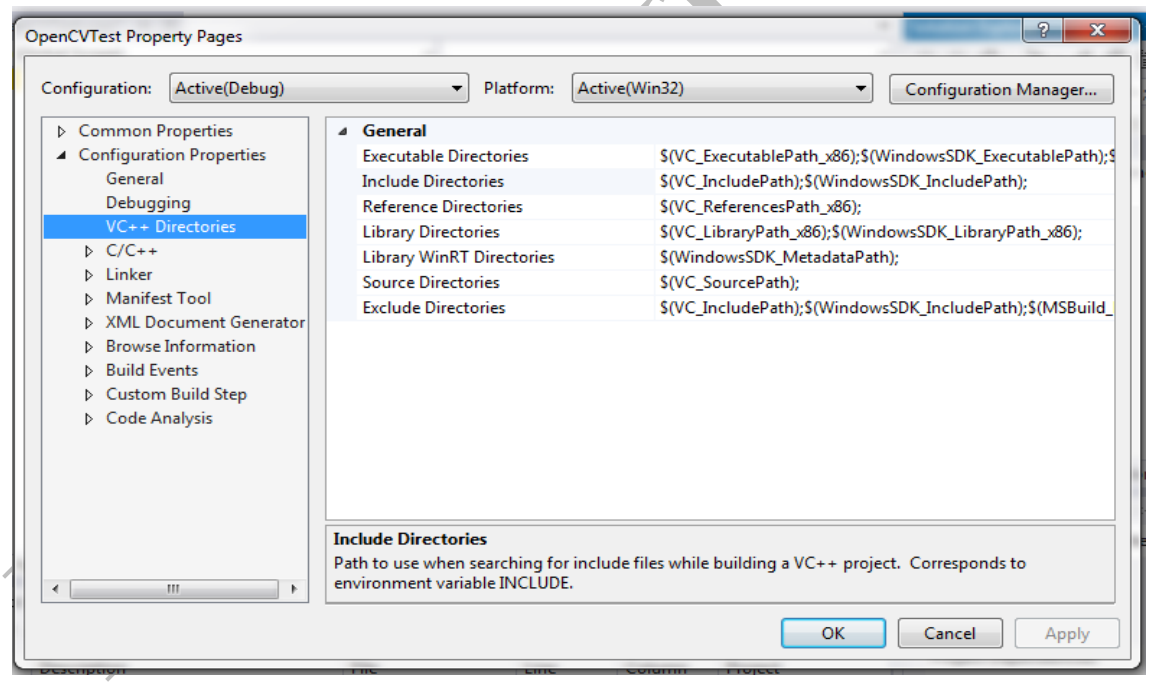
شکل (۴-۱۰) مراحل نصب

Pasnavr.com

اینک روی نام پروژه راست کلیک کنید و properties را بزنید:



شکل (۴-۱۱) مراحل نصب



شکل (۴-۱۲) مراحل نصب

سپس قسمت vc++ Directory را انتخاب کنید و در قسمت Include Directories مسیر های زیر را واردکنید:

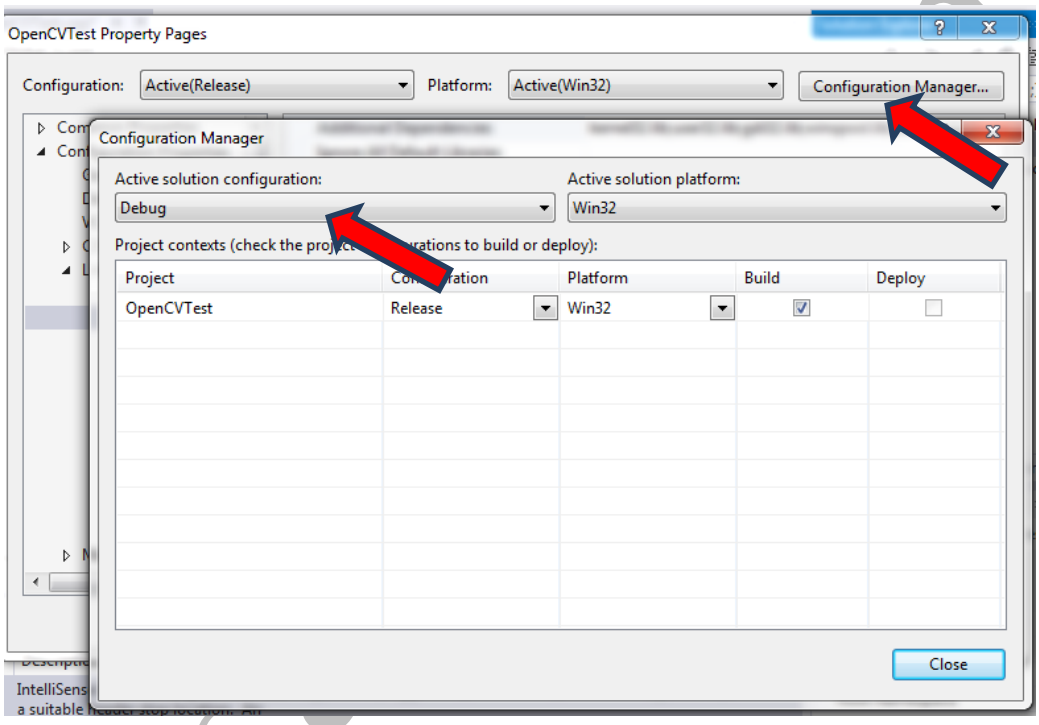
c:\opencv\build\include
C:\opencv\include\opencv
C:\opencv\include\opencv2

دوباره قسمت Library Directories را انتخاب کرده و مسیر های زیر را اضافه کنید

c:\opencv\build\x86\vc10\lib for vs2010 32bit
C:\opencv\build\x86\vc10\bin for vs2010 32bit

c:\ opencv\build\x64\vc10\lib for vs2010 64bit
C:\opencv\build\x64\vc10\bin for vs2010 64bit

سپس در قسمت Linker → Input → Additional Dependencies ابتدا configuration را روی Release بگذارید (Configuration Manager)



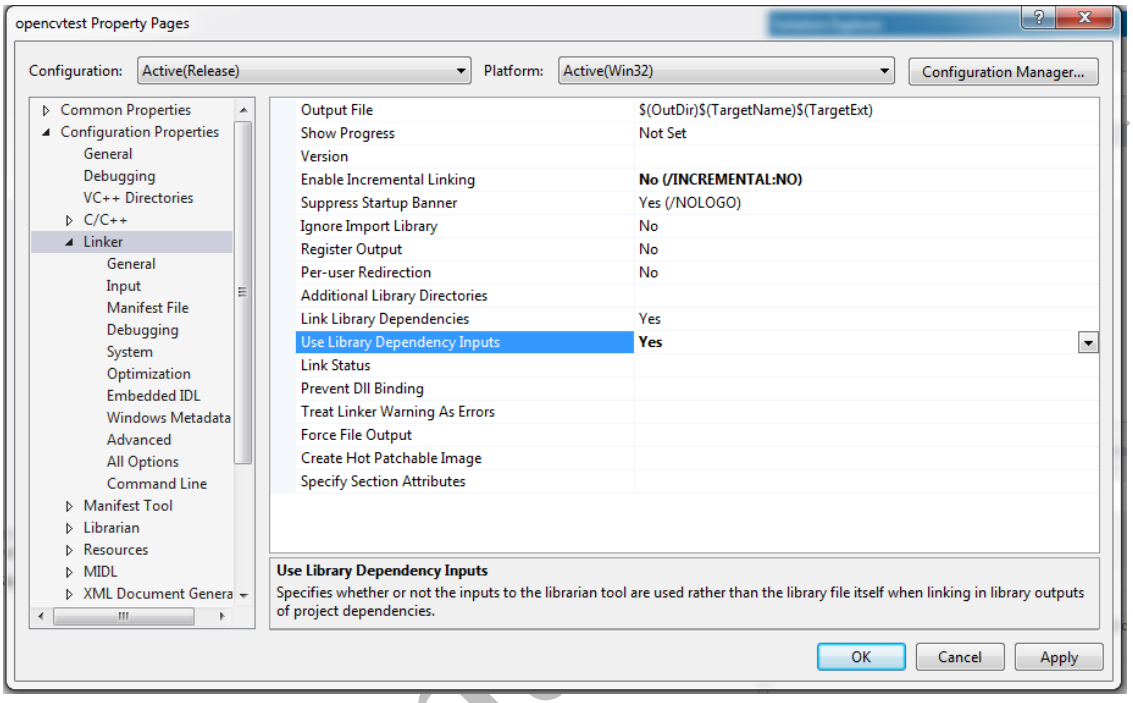
شکل (۴-۱۳) مراحل نصب

و سپس در Additional dependencies نام DLL های زیر را وارد کنید:

opencv_core246d.lib
opencv_imgproc246d.lib
opencv_highgui246d.lib
opencv_ml246d.lib
opencv_video246d.lib
opencv_features2d246d.lib
opencv_calib3d246d.lib
opencv_objdetect246d.lib
opencv_contrib246d.lib
opencv_legacy246d.lib

opencv_flann246d.lib
opencv_gpu246d.lib
opencv_nonfree246d.lib
opencv_photo246d.lib
opencv_videostab246d.lib

در قسمت Linker → Use Library Dependency Inputs را از No به Yes تغییر دهید.



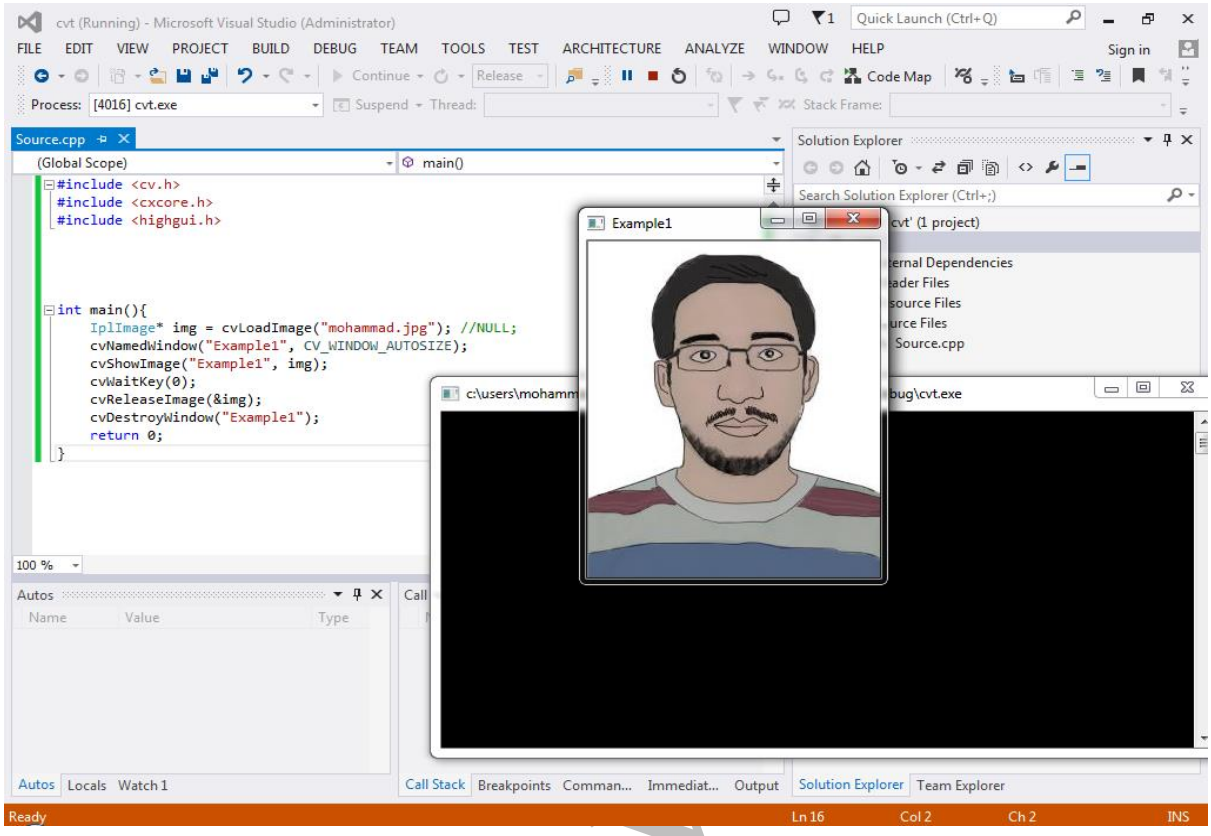
شکل (۴-۱۴) مراحل نصب

اکنون نمونه کد زیر را امتحان کنید:

```
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>

int main(){
IplImage* img = cvLoadImage("mohammad.jpg"); //NULL;
cvNamedWindow("Example1", CV_WINDOW_AUTOSIZE);
cvShowImage("Example1", img);
cvWaitKey(0);
cvReleaseImage(&img);
cvDestroyWindow("Example1");
return 0;
}
```

یک عکس به نام mohammad و با فرمت JPG به پوشه پروژه خودتان کپی کنید.

خروجی برنامه:



شکل (۴-۱۵) خروجی برنامه به زبان C++

توجه:

توضیحات مربوط به کد برنامه ، در فصول بعدی شرح داده خواهد شد.

۴-۴- نتیجه گیری

.در این فصل یاد گرفتیم که چگونه می توان opencv را در سیستم عامل ویندوز و در محیط ویژوال استدیو نصب کرد ، سپس یک برنامه ساده جهت نمایش تصویر نوشتیم.

Ehsana

فـصل ۵ :

انواع و متغیرهای اصلی در opencv

و بررسی این کتابخانه در دیگر زبان های برنامه نویسی

Ehsanavr.com

۵-۱- مقدمه

در این فصل به بررسی انواع داده و متغیر های اصلی می پردازیم و توابع پرکاربرد opencv را بررسی می کنیم تا درک مفاهیم استفاده شده در فصول بعدی ساده تر گردد.

۵-۲- انواع و متغیرهای اصلی

در opencv نام اکثر توابع و انواع با کلمه cv شروع می شوند . باید توجه داشت که ما در opencv انواعی داریم که بعضا از نام آنها به عنوان تابع نیز استفاده می شود و تفاوت اصلی بین توابع و انواع در بزرگ یا کوچک بود حرف اول آن ها می باشد. یعنی توابع با cv و انواع با Cv نوشته می شوند.

مثال:
CvSize یک نوع و cvSize یک تابع است که هردو در کتابخانه opencv موجود می باشند.

- CvPoint :

ساختاری شامل x و y از نوع صحیح (int) که قادر به نگهداری نقطه ای از تصویر است.

- CvPoint2D32f :

ساختاری شامل x و y از نوع صحیح (float) که قادر به نگهداری نقطه ای از تصویر است.

- CvPoint3D32f :

ساختاری شامل x و y وz از نوع صحیح (float) که قادر به نگهداری نقطه ای از تصویر است.

Ehsanavr.com

- CvSize :

ساختاری شامل height و width از نوع صحیح که قادر به نگهداری اندازه تصویر است.

- CvRect :

ساختاری شامل height و width از نوع صحیح که قادر به نگهداری اندازه قسمتی ازتصویر است.

- CvScalar :

آرایه ای ۴ تایی از نوع double که قادر به نگهداری عنصر تصویر یا همان رنگ پیکسل ها است.
این نوع بسیار مهم و کاربردی است . مقادیر r, b ,g, a داخل این متغیر قرار میگیرد.

نکته:
برای تمامی انواع بالا که ذکر شد ، توابعی از همان نام "با قاعده نام گذاری که گفته شد" وجود دارند که قادرند انواع فوق را بسازند. توجه کنید که نسخه اصلی opencv با زبان های C , C++ نوشته شده و به صورت ساخت یافته و شئی گرا پیاده سازی شده است و مفهوم کلاس در آنها وجود ندارد. (البته در opencvsharp که به زبان C# ویرایش شده ، کلاس هایی برای آن نوشته شده است که موضوع اصلی این پروژه نیز به زبان C# خواهد بود) لذا برای پیاده سازی opencv از struct استفاده شده است.

برای مثال:

```
Cvpoint cvpoint (int x , int y);
```

با گرفتن x و y نوعی از Cvpoint می سازد.

- CvMat :

یکی از اساسی ترین انواع در opencv است که تمامی متغیر های تصویر از این نوع مشتق شده است .
این نوع ، قابلیت ذخیره و نگهداری مقادیر مختلف داده در ۲ بُعد را داراست (حتی تصاویر و عکس ها) .
البته این نوع جزو بالاترین سطح انواع در opencv است و ما برای نگهداری تصویر از IplImage استفاده خواهیم کرد.

Ehsanavr.com

تابعی که می تواند نوع CvMat را تولید کند cvCreateMat (row, col) نام دارد.

- IplImage :

یکی از انواع اصلی opencv است .این نوع قابلیت ذخیره داده و کلیه اطلاعات مربوط به تصویر را دارا است. برخلاف قواعد نام گذاری تمام انواع ، این نوع با Ipl شروع میشود که مخفف Intel Performance Library است که مجموعه ای از دستورات به زبان C برای پردازش صوت و تصویر داراست که باعث افزایش سرعت در پردازش می شود.

توجه:
در صورت خرید کتابخانه Ipp اینتل که قبلا توضیح داده شد . در برنامه نویسی بجای نوشتن IplImage می توان از IppImage استفاده کرد که سرعت پردازش تصویر را افزایش خواهد داد.

۵-۲-۱- انواع داده در تصویر

در متغیر های IplImage داده هایی از نوع تصویر هم ذخیره می شوند و نوع این داده ها ممکن است متفاوت باشند و این تفاوت در قالب جدولی در زیر آمده است.

جدول (۵-۱) انواع داده در تصویر

Unsigned 8 bit integer (8u)	*IPL_DEPTH_8U :*
Signed 8 bit integer (8s)	*IPL_DEPTH_8S :*
Signed 16 bit integer (16s)	*IPL_DEPTH_16S :*
Signed 32 bit integer (32s)	*IPL_DEPTH_32S :*
32-bit fl oating-point single-precision (32f)	*IPL_DEPTH_32F :*
64-bit fl oating-point double-precision (64f)	*IPL_DEPTH_64F :*

۵-۲-۲- دسترسی به عناصر تصویر با cvGet2D و cvSet2D

جهت دسترسی به مقادیر مختلف رنگ در تصویر از دستور cvGet2D استفاده می شود.
و همچنین cvSet2D جهت تعیین مقادیر هر پیکسل می باشد.

avr.com

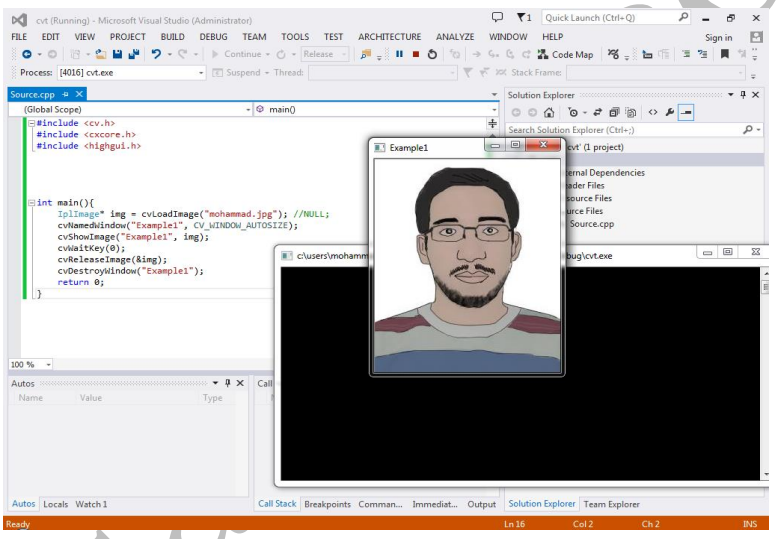
نکته:

در صورتی که تصویر بیش از ۲ بُعد داشته باشد می توان از cvGetND و cvSetND استفاده کرد.

```
void cvSet2D( CvArr* arr, int idx0, int idx1, CvScalar value );
void cvSetND( CvArr* arr, int* idx, CvScalar value );
```

۵-۲-۳- توضیحات مربوط به کد اولیه برنامه (نمایش یک تصویر)

در این قسمت قصد داریم تا در قالب مثل کاربردی ، برنامه ای را که در فصل قبل نوشتیم را توضیح دهیم تا در عمل با مفهوم توابع و انواع آشنا شویم .



شکل (۵-۱) مثال کاربردی

خواندن تصاویر در opencv و نمایش آنها ، طی سه مرحله انجام میشود.

1. **ساخت یک پنجره**
2. **بارگذاری تصویر**
3. **نمایش تصویر**

(البته ترتیب مراحل اول و دوم مهم نیست)

Ehsa.com

۱. ساخت یک پنجره

برای نمایش عکس نیاز است که یک پنجره جهت قرار گرفتن تصویر در آن بسازید این کار با استفاده از دستور cvNamedWindow انجام میشود.

```
cvNamedWindow("Example1", CV_WINDOW_AUTOSIZE);
```

آرگومان اول این دستور ، رشته ای است که شامل نام دلخواه پنجره می شود و آرگومان دوم نحوه نمایش پنجره را توصیف می کند.

آرگومان دوم می تواند به صورت های زیر نیز باشد.

CV_WINDOW_NORMAL
CV_WINDOW_AUTOSIZE
CV_WINDOW_FULLSCREEN

۲. بارگذاری تصویر

خواندن و بارگذاری تصویر با دستور cvloadImage انجام میشود.آرگومان های این تابع یک رشته است که شامل مسیر ورودی عکس میشود.

```
IplImage* img = cvLoadImage("mohammad.jpg");
```

در کد بالا "mohammad.jpg" مشخص می کند که فایلی با همین نام را از میسر جاری پوشه برنامه نویسی load کند.

این تابع یک آرگومان عددی (int) به عنوان پارامتر دوم نیز میگیرد که نشان دهنده سیاه-سفید و یا رنگی بودن تصویر بارگذاری شده است.

```
IplImage* img = cvLoadImage("mohammad.jpg" , 1);
```

عدد یک به معنای قرار دادن تصویر به صورت رنگی است.

۳. نمایش تصویر

برای نمایش تصویر از دستور cvShowImage استفاده میکنیم .آرگومان اول این دستور شامل یک رشته است و حاوی نام پنجره ای است که می خواهیم تصویر در آن نمایش داده شود و آرگمان دوم متغیری است که تصویر در آن قرار دارد و در مثال ما با IplImage و با نام img ساخته شده است .

```
cvShowImage("Example1", img);
```

از بقیه دستوارات جهت نگه داشتن تصویر نمایش داده شد در برنامه و از بین بردن آن استفاده میشود.

```
cvWaitKey(0);
cvReleaseImage(&img);
cvDestroyWindow("Example1");
```

Ehsanavi.com

دستور اول منتظر ورود کلیدی از کیبورد جهت خاتمه دادن به برنامه توسط کاربر می ماند.
دو دستور بعد تصویر را از حافظه ram خالی کرده و پنجره ساخته شده را از بین می برد.
البته دستور cvWaitKey(0) می تواند کاربردهای دیگری نیز داشته باشد ، مثلا با دادن عدد به جای صفر می تواند برنامه را تا زمان عدد وارد شده (به میلی ثانیه) بالا نگه داشت.
یقیه دستورات opencv که بسیار کاربردی نیز هستند در فایل های ضمیمه موجود بوده و پیوست خواهند شد ولی بررسی تک تک آنها از حوصله خواننده خارج است.
تا اینجا به معرفی و بررسی کتابخانه opencv به زبان C++ به جهت معرفی پرداختیم .
در ادامه برنامه نویسی با opencv را به زبان برنامه نویسی C# انجام خواهیم داد و پیاده سازی پروژه را در آن زبان بررسی خواهیم کرد.

۵-۳- بررسی کتابخانه opencv در دیگر زبان های برنامه نویسی و سطح بالا

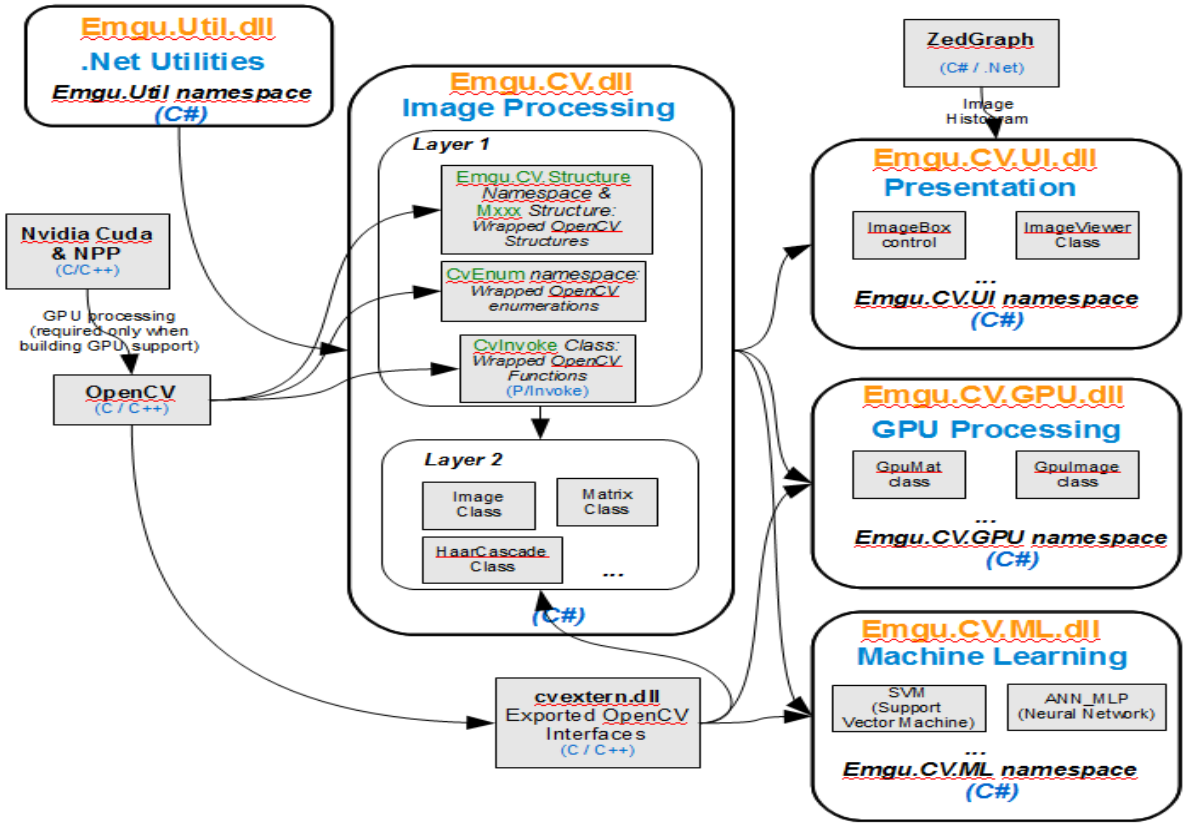
از آنجا که دراین کتابخانه از مفهوم کلاس برای پیاده سازی ها استفاده نشده بود لذا پیاده سازی این کتابخانه و استفاده از آن در زبان های سطح بالا کاری بسیار مشکل بود .
بنابراین گروهی از برنامه نویسان مستقل ، اقدام به پیاده سازی این کتابخانه پردازش تصویر در دیگر زبان های برنامه نویسی کردند.
مثلا برای زبان برنامه نویسی java کتابخانه ای تحت عنوان javacv مطرح شد که توسعه داده شده ی کتابخانه opencv است .

در زبان برنامه نویسی C# هم دو کتابخانه مطرح هستند که یکی از آنها opencvsharp نام دارد و دیگری EmguCv نام دارد . هردو زبان تمامی قواعد opencv را به همان شکلی که در فصل قبل مطرح شد ، پیاده سازی کرده اند اما Emgucv از namespace های دلخواه خود استفاده کرده است وکلاس بندی اجزای opencv را طبق قوانین خود انجام داده است .

بنابراین در صورت کار با Emgucv نیاز است که به منابع آن در سایتش مراجعه کرد.

Ehsanavr.com

آدرس مربوط به آن **http://www.emgu.com/wiki/index.php** است و تمامی توضیحات آن جامع و کامل است . ما در این پروژه از Emgucv استفاده نخواهیم کرد بنابراین به نمایش یک تصویر از آن این کتابخانه سطح بالا بسنده می کنیم.



شکل (۵-۲) کتابخانه EmguCV

اما در رابطه با opencvsharp ، این کتابخانه کاملا منطبق با opencv است و ابتدای شروع آن توسط گوگل کلید خورد و در google code افراد به صورت آزادانه اقدام به توسعه آن می کردند اما با گذشت چند نسخه از آن ، کل پروژه به سایت github منتقل شد.

آدرس این کتابخانه به صورت :

https://github.com/shimat/opencvsharp است و نکته جالب این کتابخانه این است که در تمامی زبان های مشتق شده از سی ، شامل C و C++ و C# قابل استفاده است و نحوه استفاده از آن بسیار راحت و مطابق با قوانین توابع و انواع درopencv است .

بسیاری از توابع را می توان با استفاده از اضافه کردن یک ***Cv*** به ابتدای آن بکار برد.

مثال :

```
Cv.ShowImage("src image", src);
```

البته ناگفته نماند که باید با استفاده از using ، کتابخانه opencvsharp را در برنامه فرخوانی کنید.

```
using OpenCvSharp;
```

نحوه اضافه کردن این کتابخانه به پروژه شما بسیار ساده است وتنها کافیست چند dll را به پروژه خود import کنید.

تنها مشکل این کتابخانه (opencvsharp) این است که ، چون به زبان سطح بالا پیاده سازی و کلاس بندی شده است ، به مراتب از سرعت کمتری نسبت به پدر خود یعنی OpenCv برخوردار است ولی نحوه استفاده آن بسیار راحتر و مطبوع تر است.

مثالی از برنامه نوشته شده ی موجود در سایت opencvsharp

```
using OpenCvSharp;

class Program
{
    static void Main()
    {
        IplImage src = Cv.LoadImage("lenna.png", LoadMode.GrayScale);
        IplImage dst = Cv.CreateImage(new CvSize(src.Width, src.Height),
BitDepth.U8, 1);
        Cv.Canny(src, dst, 50, 200);// فیلتر گذاری بر روی تصویر
        Cv.NamedWindow("src image");
        Cv.ShowImage("src image", src);
        Cv.NamedWindow("dst image");
        Cv.ShowImage("dst image", dst);
        Cv.WaitKey();
        Cv.DestroyAllWindows();
        Cv.ReleaseImage(src);
        Cv.ReleaseImage(dst);
    }
}
```

متاسفانه منابع بسیار محدودی (تقریبا هیچ) در سایت های فارسی زبان ، مربوط به opencvsharp موجود است.

۵-۴- نتیجه گیری

در این فصل متغیر های ضروری کتابخانه opencv را جهت شروع برنامه نویسی بررسی کردیم . بدیهی است که بررسی تمامی ساختار ها و توابع آن از حوصله خوانندگان گرامی خارج است.

سعی بر این است که یک خلاصه از تمامی توابع را به صورت انگلیسی در پیوست این تحقیق ضمیمه کنیم.

فـصل ۶ :

استفاده از Opencv در یک Windows Form Application

در ویژوال استدیو به زبان C#

Ehsanavr.com

۶-۱- مقدمه

در این فصل به شروع برنامه نویسی با زبان سطح بالای C# می پردازیم و نحوه استفاده از سخت افزارهای خارجی مانند دوربین (وبکم) را فرا خواهیم گرفت .
سپس اقدام به هنگام سازی کتابخانه opencv با تصویر گرفته شده از دوربین و تبدیل آن به یک تصویر استاندارد جهت نمایش در یک picture box خواهیم کرد.

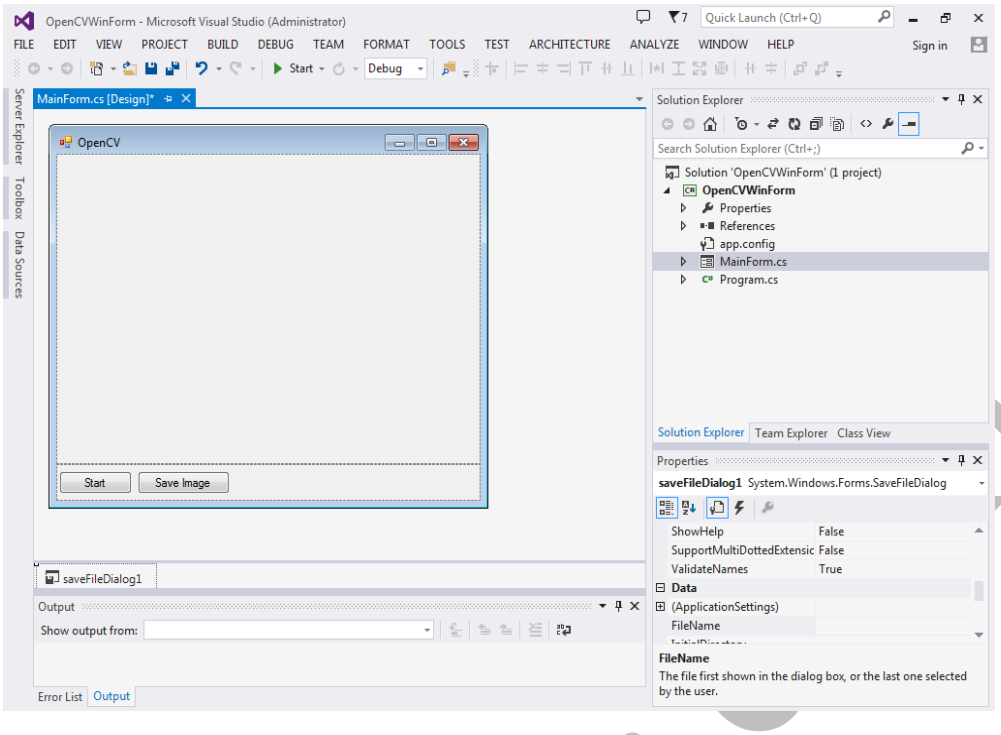
تا اینجا از توابع داخلی opencv جهت نمایش تصویر استفاده می کردیم . اما این کار زیبایی و انسجام برنامه نوشته شده را از بین می برد ، بنابراین استفاده از ابزار های ماکروسافت ویژوال استودیو راه بهتری برای نمایش تصاویر در قالب فرم های ویندوزی است.

شکل (۶-۱) نمایش تصویر در Windows form

۶-۲- ایجاد پروژه و طراحی فرم

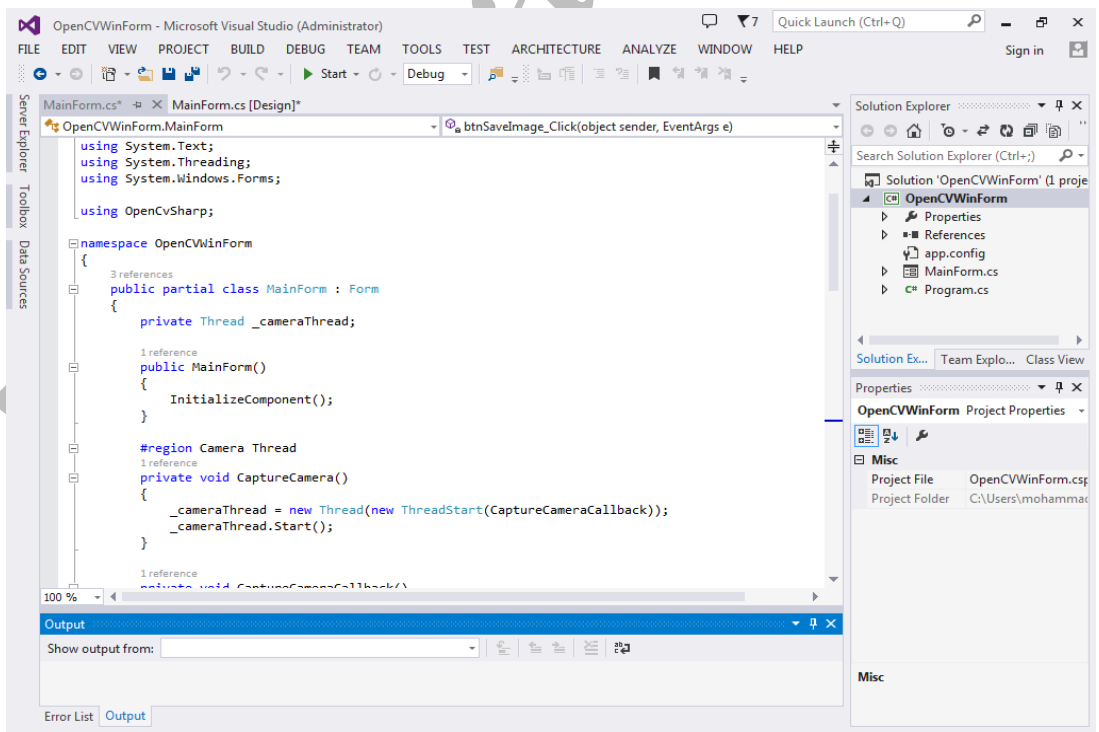
برای این کار ابتدا پروژه ای از نوع Windows Form Aplication در محیط ویژوال استدیو ساخته و سپس اقدام به طراحی فرم مورد نظر می کنم .
در این فرم از دو عدد Button با نام های start و save image استفاده شده است.
همچنین یک picture box و یک saveFileDialog که از منوی کشویی ToolBox به فرم اضافه شده است .
پارامتر name دو button را به ترتیب به btnSaveImage و btnStart تغییر داده ایم تا در کد زدن راحت تر عمل کنیم.

Ehsan.com



شکل (۶-۲) طراحی فرم

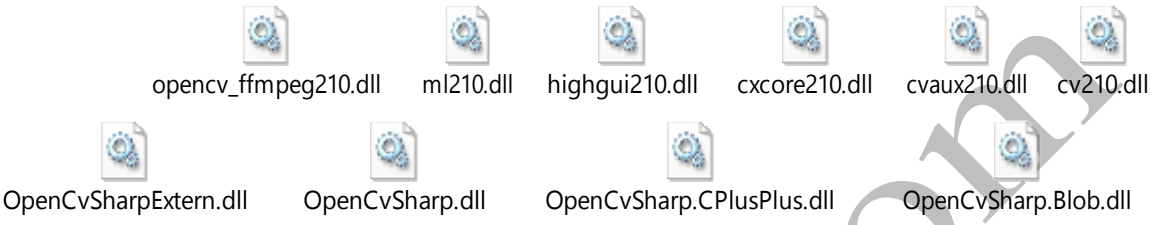
بر روی فرم دوبار کلیک کرده تا وارد محیط برنامه نویسی آن شویم .



شکل (۶-۳) محیط کد زنی

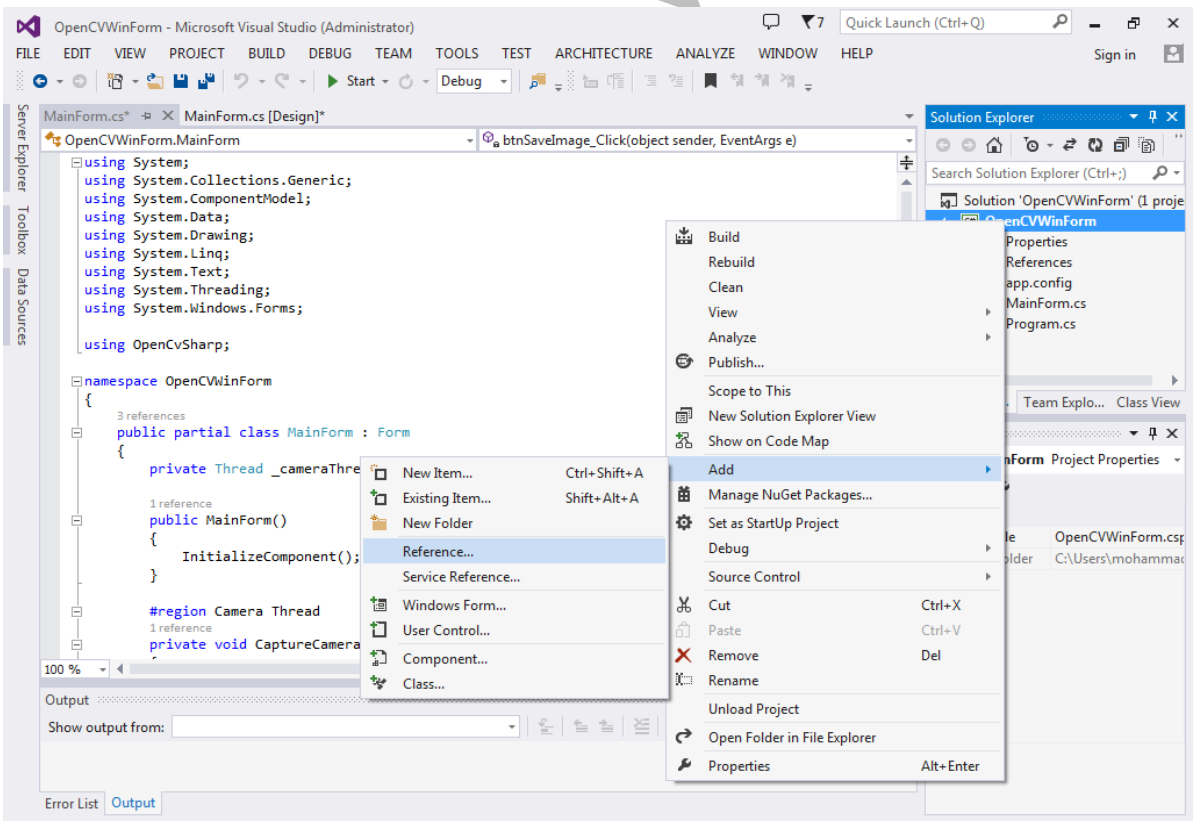
۶-۳- اضافه کردن dll های مربوطه (opencvsharp)

در پروژه باید dll های مربوط به opencvsharp را اضافه کنیم.
این dll ها در سایتی که در بالا معرفی شد موجود بوده و همراه پروژه نیز ضمیمه میگردد.
این dll ها عبارتند از :



نکته مهم این است که در بیشتر موارد فقط اضافه کردن openCvSharp به پروژه کافی است.

در solution Explorer بر روی نام پروژه راست کلیک کرده و گزینه Add و سپس Reference را زده و dll های مربوطه را اضافه می کنیم.



شکل (۶-۴) اضافه کردن Dll های مربوطه

پس از اضافه کردن Dll می توان کد زدن را در پروژه شروع کرد.

۶-۴- شروع برنامه نویسی

در پروژه نیاز است که حتما دو مورد using انجام داد.

```
using OpenCvSharp;
using System.Threading;
```

اولین using جهت استفاده از کتابخانه پردازش تصویر است و دومین using جهت استفاده از thread به منظور فراخوانی دوربین می باشد.

نکته:

در صورت عدم استفاده از thread برای دوربین ممکن است برنامه crash کند و یا روند اجرای برنامه به مشکل برخورد کند زیرا بکارگیری دوربین و پردازش فرم برنامه با هم در یک thread امکان پذیر نیست.
توصیه این است که هرگاه برنامه مجاب به استفاده از سخت افزار و یا ورودی خاصی است ، همواره سعی شود که از threading در برنامه استفاده شود.
پس از using ها در کلاس فرم اصلی ، یک متغیر از نوع thread میسازیم.

```
private Thread _cameraThread;
```

تا بتوانیم با آن دوربین را فراخوانی کنیم .
سپس یک تابع جهت فراخوانی یک تابع دیگر به صورت thread و موازی با روند اصلی برنامه میسازیم.

```
private void CaptureCamera()
{
    _cameraThread = new Thread(new ThreadStart(CaptureCameraCallback));
    _cameraThread.Start();
}
```

تابع CaptureCamera یک تابع با نام CaptureCameraCallback را به صورت موازی با روند اصلی برنامه صدا زده و آن را استارت میکند و وظیفه آن فریم بندی تصویر و نمایش آن در فرم است.
و اما تابع CaptureCameraCallback() :

```
private void CaptureCameraCallback()
{
    using (CvCapture cap = CvCapture.FromCamera(CaptureDevice.Any, 1))
    {
        while (CvWindow.WaitKey(10) < 0)
        {
            Bitmap bm = BitmapConverter.ToBitmap(cap.QueryFrame());
            bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height);
            pctCvWindow.Image = bm;
        }
    }
}
```

در خط اول این برنامه متغیر cap اقدام به گرفتن تصویر از دوربین میکند .
CaptureDevice.Any مشخصه دوربین بوده و عدد یک شماره سخت افزار را مشخص میکند و معمولا در وبکم های داخلی لب تاپ و دوربین های جلوی تلفن همراه این عدد از یک به صفر باید تغییر کند.
دستور while برنامه را مجاب می کند که درون حلقه را بارها اجرا کند.
تابع waitkey() متنظر گرفتن ورودی از کاربر میشود و پس از گرفتن کلید ورودی از کاربر ۱۰ میلی ثانیه صبر میکند و سپس مقدار کد اَسکی آن کلید ورودی را برمیگرداند اما از آنجایی که در ظاهر فرم مقداری برای ورود کلید تعبیه نشده است پس CvWindow.WaitKey(10) همیشه مقدار -1 را برمیگرداند. و میتوان شرط داخل حلقه را به while(true) تغییر داد.

در درون حلقه ما با cap.QueryFrame() تصویر ورودی دوربین را frame بندی می کنیم یعنی بسته به نوع دوربین مقداری بین ۱۵ تا ۳۰ فریم در ثانیه از دوربین گرفته می شود و هر فریم تصویری را تبدیل به فرمت Bitmap (نوعی فرمت تصویری bmp.) می کند . علت این کار این است که ما می خواهیم این تصویر را بر روی picture box موجود در فرم قرار دهیم و این picture box فقط فرمت های استاندارد تصویر را قبول می کند.

نکته:
برای استفاده از کلمه کلیدی Bitmap باید از using زیر استفاده کنیم.

```
using System.Drawing;
```

پس از آن با خط bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height); تصویر موجود را به اندازه طولی و عرضی picture box در می آوریم.
pctCvWindow نام picture box مورد نظر ما در این پروژه است.
در نهایت تصویر موجود را در picture box قرار می دهیم و این کار با استفاده از دستور

pctCvWindow.Image = bm;

انجام می شود.

اما تابع بعدی مورد بحث جهت آزاد سازی thread استفاده شده است و منابع سیستمی را به هنگام بستن برنامه و فرم ، آزاد میکند و نیازی به توضیح ندارد.

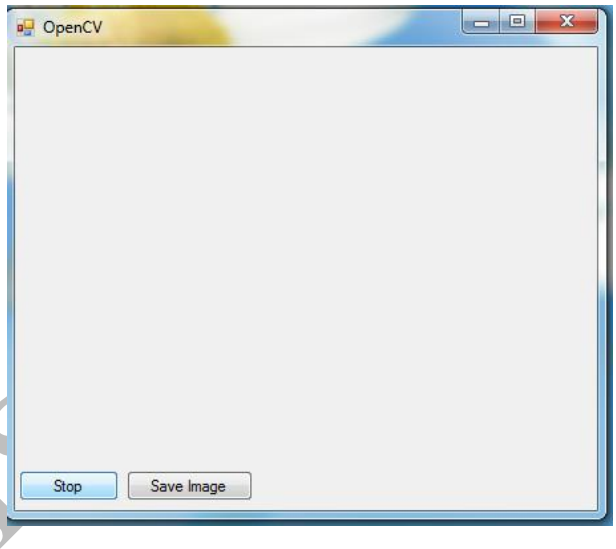
```
private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    if (_cameraThread != null && _cameraThread.IsAlive)
    {
        _cameraThread.Abort();
    }
```

اکنون نوبت به برنامه نویسی button های استفاده شده در برنامه است.

در فرم طراحی شده بر روی دکمه start دوبار کلیک کنید تا وارد محیط برنامه نویسی مربوط به این دکمه شویم.

```
private void btnStart_Click(object sender, EventArgs e)
{
    if (btnStart.Text.Equals("Start"))
    {
        CaptureCamera();
        btnStart.Text = "Stop";
    }
    else
    {
        _cameraThread.Abort();
        btnStart.Text = "Start";
    }
}
```

در تابع مربوط به دکمه استارت یک شرط داریم که میگوید در صورت زدن دکمه start توسط کاربر ابتدا تابع مربوط به فراخونی دوربین ، که با thread شروع می شود را فراخوانی کند و سپس نام روی دکمه را به stop تغییر دهد. و در غیر اینصورت (یعنی با زدن دکمه stop) فراخوانی دوربین متوقف می شود.

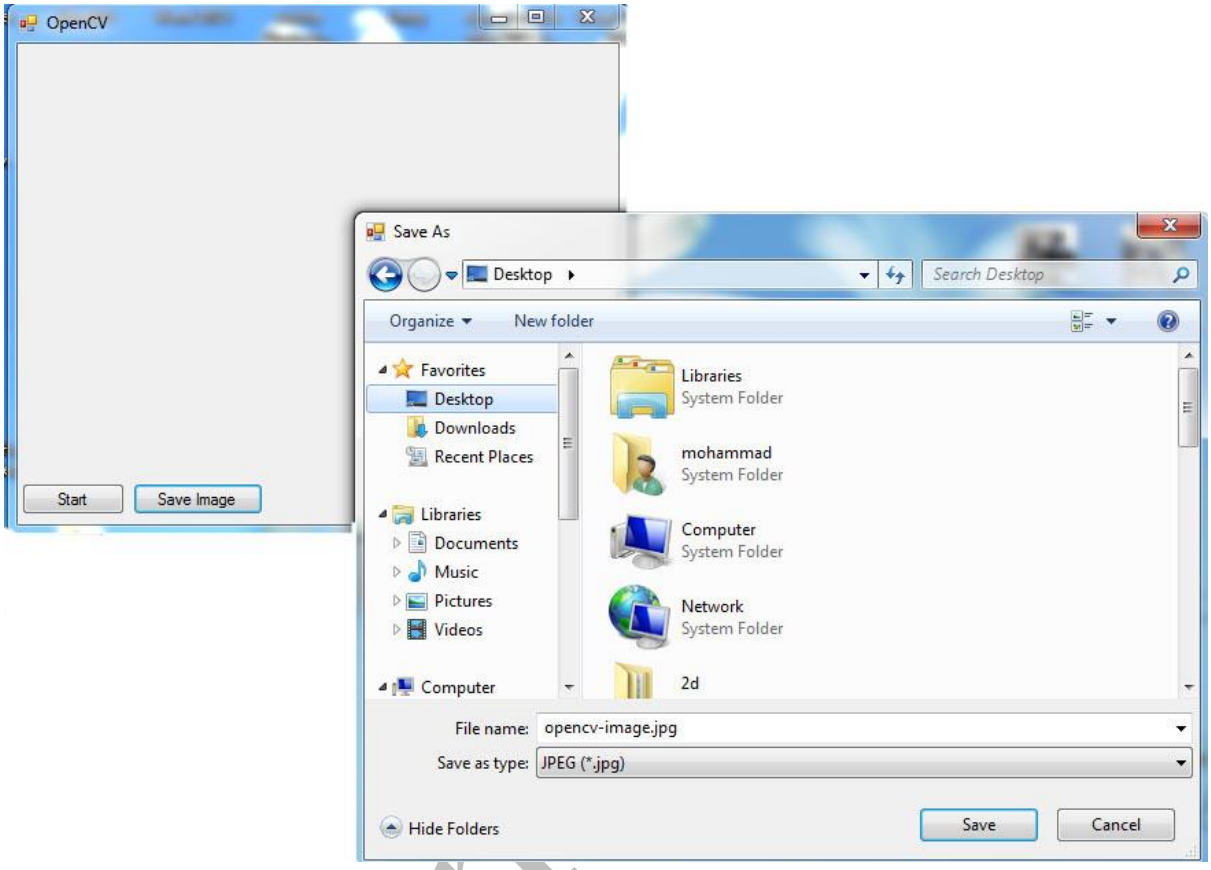


شکل (۶-۵) دکمه Stop در فرم

برای ذخیره سازی تصویر مثل روند استارت ، ابتدا روی دکمه save image دوبار کلیک کنید و سپس دستوارت زیر را بکار ببرید.

```
private void btnSaveImage_Click(object sender, EventArgs e)
{
    saveFileDialog1.FileName = "opencv-image.jpg";
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        String imageFileName = saveFileDialog1.FileName;
        pctCvWindow.Image.Save(imageFileName);
    }
}
```

این دستورات از SaveFileDialog موجود در ویژوال استدیو جهت ذخیره سازی تصویر در محل دلخواه شما و با نام دلخواه شما استفاده می کند و تصویر موجود در picture box را ذخیره می کند.



شکل (۶-۶) ذخیره سازی تصویر

و این بود تمامی مراحل مورد نیاز جهت نمایش و ذخیره تصاویر دوربین در یک WindowsForm در #C.

۶-۵- نتیجه گیری

از مهمترین فعالیت های لازم برای پردازش تصویر خواندن تصویر از ورودی و نگاشت آن در مکان مورد نظر است که در این فصل در قالب مثالی کاربردی آن را پیاده سازی کردیم و کد های مورد نظر را یک به یک شرح دادیم.

فـصل ۷ :

تطبیق الگو با استفاده از opencv در

Windows Form Application C#

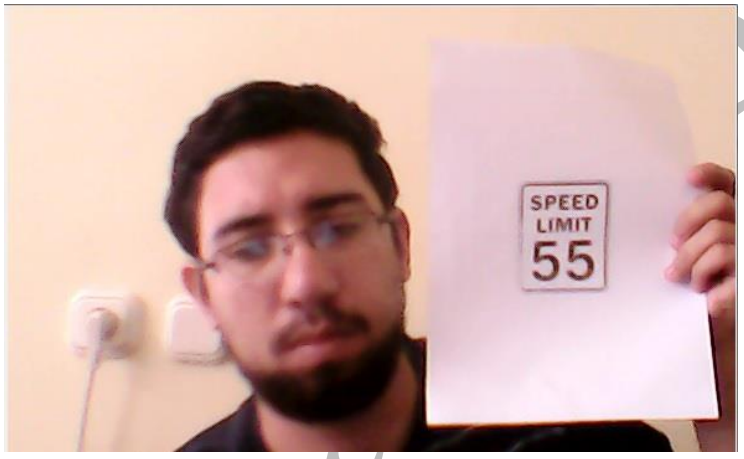
Ehsanavar.com

۷-۱- مقدمه

تطبیق الگو یکی از زمینه های کاربردی و مهم در پردازش تصویر است که استفاده و کاربردهای گوناگونی در صنعت و ابزارهای روزمره زندگی دارد در این مثال کاربردی ما قصد داریم که نمونه یک برنامه تشخیص علائم راهنمایی و رانندگی را بنویسیم البته این مورد یک مثال پایه ای بوده و نیاز به کار بیشتری بر روی آن است.

در این مثال ما قصد داریم که کامپیوتر برای ما تابلویی با علامت محدودیت سرعت ۵۵ کیلومتر را تشخیص دهد.

من یک طرح اولیه از این تابلو را بر روی کاغذ A4 پرینت نموده ام .



شکل (۷-۱) محدودیت سرعت ۵۵ کیلومتر

ما می خواهم برنامه ای بنویسم که در صورت تشخیص این تابلو ، کادر سبز رنگی دور آن نمایان کند.

یعنی:

شکل (۷-۲) تطبیق الگو

Ehsan

۷-۲- توضیحات مربوط به کد و برنامه نویسی

مانند مثال فصل قبل یک پروژه جدید از نوع Windows Form Applicatin C# ایجاد می کنیم تمامی مراحل کار مانند مثال فصل قبلی است با این تفاوت که تغییرات اندکی در تابع CaptureCameraCallback() ایجاد می کنیم .

نکته مهمی که وجود دارد این است که ما باید از قبل مشخصات تابلو را در قالب یک عکس به برنامه بدهیم تا برنامه با الگوریتم های تطبیق الگو هر یک از فریم های دوربین را بررسی کند و در صورت پیدا کردن الگو مختصات آن را در تصویر ورودی ثبت و اقدام به کشیدن کادر سبز رنگ دور آن بکند.

بنابراین ابتدا عکس تابلو را در پوشه bin/Debug مربوط به پروژه باید قرار دهید.

شکل (۷-۳) تابلو

سپس مراحل کد زنی مورد نیاز را بررسی می کنیم.

مانند مثال فصل قبل که جهت نمایش تصویر در یک ویندوز فرم بررسی می شد تمامی مراحل مشابه است ولی فقط یک تابع را تغییر می دهیم و آن تابع ، تابع زیر است :

```
        private void CaptureCameraCallback()
        {
            CvCapture cap = CvCapture.FromCamera(1);
            IplImage tpl = new IplImage("speedlimit55.jpg", LoadMode.AnyColor |
LoadMode.AnyDepth);
            CvPoint minloc;
            CvPoint maxloc;
            double minval;
            double maxval;
            double thresh = 0.3;
            while (true)
                {
                    IplImage img = cap.QueryFrame();
                    IplImage res = Cv.CreateImage(Cv.Size(img.Width - tpl.Width + 1,
img.Height - tpl.Height + 1), BitDepth.F32, 1);
                    IplImage gimg = Cv.CreateImage(Cv.GetSize(img), img.Depth, 1);
```

```
                    IplImage gtpl = Cv.CreateImage(Cv.GetSize(tpl), img.Depth, 1);
                    Cv.CvtColor(img, gimg, ColorConversion.BgrToGray);
                    Cv.CvtColor(tpl, gtpl, ColorConversion.BgrToGray);
                    Cv.MatchTemplate(gimg, gtpl, res,
MatchTemplateMethod.CCoeffNormed);
                    Cv.MinMaxLoc(res, out minval, out maxval, out minloc, out maxloc,
null);
                    if (maxval > thresh)
                    {
                        Cv.Rectangle(img, Cv.Point(minloc.X + tpl.Width, minloc.Y +
tpl.Height), Cv.Point(minloc.X, minloc.Y), CvColor.Green, 5, 0, 0);
                    }
                    Bitmap bm = BitmapConverter.ToBitmap(img);
                    bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height);
                    pctCvWindow.Image = bm;
                }
```

(تمامی پروژه ها و مثال ها در قالب dvd ضمیمه خواهند شد.)

و اما توضیحات :

در خط اول این تابع ورودی دوربین مشخص می شود تا بتوان به دوربین در برنامه دسترسی مستقیم داشته باشیم.

در خط دوم کد

```
IplImage tpl = new IplImage("speedlimit55.jpg", LoadMode.AnyColor |
LoadMode.AnyDepth);
```

تصویر تابلو را که با نام speedlimit55.jpg در پوشه bin/Debug قرار دادیم را در برنامه فراخوانی میکنیم .

سپس تعریف پنج متغیر را داریم :

```
CvPoint minloc;
CvPoint maxloc;
double minval;
double maxval;
double thresh = 0.3;
```

دو متغیر اول نگهدارنده مختصات مکانی اند و دو مقدار مینیمم و ماکزیمم را که شامل x و y تصویر است را در هرکدام از متغیر ها خواهیم ریخت.

دومقدار minval و maxval دو عدد داده آماری را که از الگوریتم تطبیق الگو دریافت می کنند ، در خود نگه میدارند. مقدار thresh نیز مقداری دلخواه بوده که برای نمایش کادر سبز استفاده میشود . به طور کلی استفاده از این متغیر ها درمتن برنامه روشن تر خواهد شد.

اکنون به یک حلقه بینهایت while (true) میرسیم که تک تک فریم های دریافتی از دوربین را برای تطبیق با الگوی خودش (speedlimit55.jpg) در هر لحظه بررسی می کند.

مهمترین بخش این تابع فعالیت هایی است که در داخل این حلقه انجام می شود.

با اولین دستور داخل حلقه یعنی `IplImage img = cap.QueryFrame();` ابتدا تصویر گرفته شده از دوربین را فریم بندی می کنیم تا بتوان به اطلاعات تصویر در هر لحظه دسترسی داشت.

rasanavr.com

دستور:

```
IplImage res = Cv.CreateImage(Cv.Size(img.Width - tpl.Width + 1, img.Height - tpl.Height + 1), BitDepth.F32, 1);
```

یک حافظه جهت نگهداری داده های حاصل الگوریتم تطبیق الگو را ایجاد می کند . ما آن را در قالب یک تصویر در نظر میگیریم زیرا شامل داده های تصویری نیز می باشد .اما قابل نمایش نیست و فقط داده های آماری در آن ذخیره میشود . طول و عرض این داده های آماری از تفریق حاصل از تصویر فریم بندی شده دوربین وعکس تابلو بزرگتر نیست . عدد یک آخر این دستور (قبل از بستن پرانتز) نشان دهنده تعداد کانال های تصویر است.

اگر می خواهیم داده های ذخیره شده در متغییر res رنگی باشد کافیست عدد ۱ را به ۳ تغییر دهیم بنابراین عدد ۱ نشان دهنده سیاه-سفید بودن نوع تصویر است.

دو خط بعدی

```
IplImage gimg = Cv.CreateImage(Cv.GetSize(img), img.Depth, 1);
IplImage gtpl = Cv.CreateImage(Cv.GetSize(tpl), img.Depth, 1);
```

دو عدد حافظه تصویری به اندازه تصویر دوربین و تصویر تابلو به منظور نگهداری تصاویر grayscale ایجاد می کند.

در دوخط بعد ما با دستورات زیر مقدار img و tpl را سیاه-سفید کرده و به ترتیب درgimg وgtpl قرار می دهیم . علت اینکه ما کپی سیاه-سفید از دوربین و تابلو محدودیت سرعت تهیه کردیم این است که این دو مقدار را به عنوان ورودی به الگوریتم تطبیق الگو بدهیم تا در روند پردازش و تطبیق الگو سرعت بیشتری (نسبت به تطبیق تصاویر رنگی) داشته باشیم.

سپس میرسیم به خط مربوط به فراخوانی الگوریتم تطبیق الگو یعنی:

```
Cv.MatchTemplate(gimg, gtpl, res, MatchTemplateMethod.CCoeffNormed);
```

پارامتر اول این تابع gimg یعنی تصویر سیاه-سفید دوربین و پارامتر دوم gtpl تصویر تابلو محدودیت سرعت است. پارامتر سوم مکانی است که خروجی حاصل از تابع MatchTemplate در آن قرار میگیرد که شامل داده های آماری است . و پارامتر سوم نوع الگوریتم استفاده شده جهت تطبیق الگو را نشان می دهد.

روش کلی این الگوریتم این است که بر حسب میزان نزدیکی پیکسل ها به الگوی مورد نظر داده های آماری را در res ذخیره می کند و این داده ها با خط بعدی قابل خواندن هستند یعنی:

```
Cv.MinMaxLoc(res, out minval, out maxval, out minloc, out maxloc, null);
```

این خط بیشترین مقدار شباهت و کمترین مقدار شباهت را که معمولا به صورت مقدار اعشاری است از پارامتر اول تابع یعنی res خوانده و در minval و maxval ذخیره می کند. علت استفاده از کلمه کلیدی

out این است که از آنجایی که تابع Cv.MinMaxLoc مقدار بازگشتی ندارد ، لذا کلمه کلیدی out متغییرها را به تابع معرفی می کند تا تغییرات را به متغیر معرفی شده به عنوان پارامتر ارسال نماید.
البته می توان از کلمه کلیدی ref نیز استفاده کرد ولی در صورت استفاده از این کلمه کلیدی نیاز است که متغیرها از پیش مقدار داشته باشند.

دو پارامتر بعدی minloc و maxloc مختصات الگوی تطبیق یافته در تصویر را درخود ذخیره می کند.
پارامتر آخر نوع ماسک تصویری است که در این مثال کاربردی ندارد.
جهت اطلاعات بیشتر میتوان به آدرس زیر جهت کارکرد الگوریتم تطبیق الگو مراجعه کرد.

http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html

البته مثال های این صفحه به صورت کدهای C++ است.
می رسیم به بررسی شرط if موجود در کد:

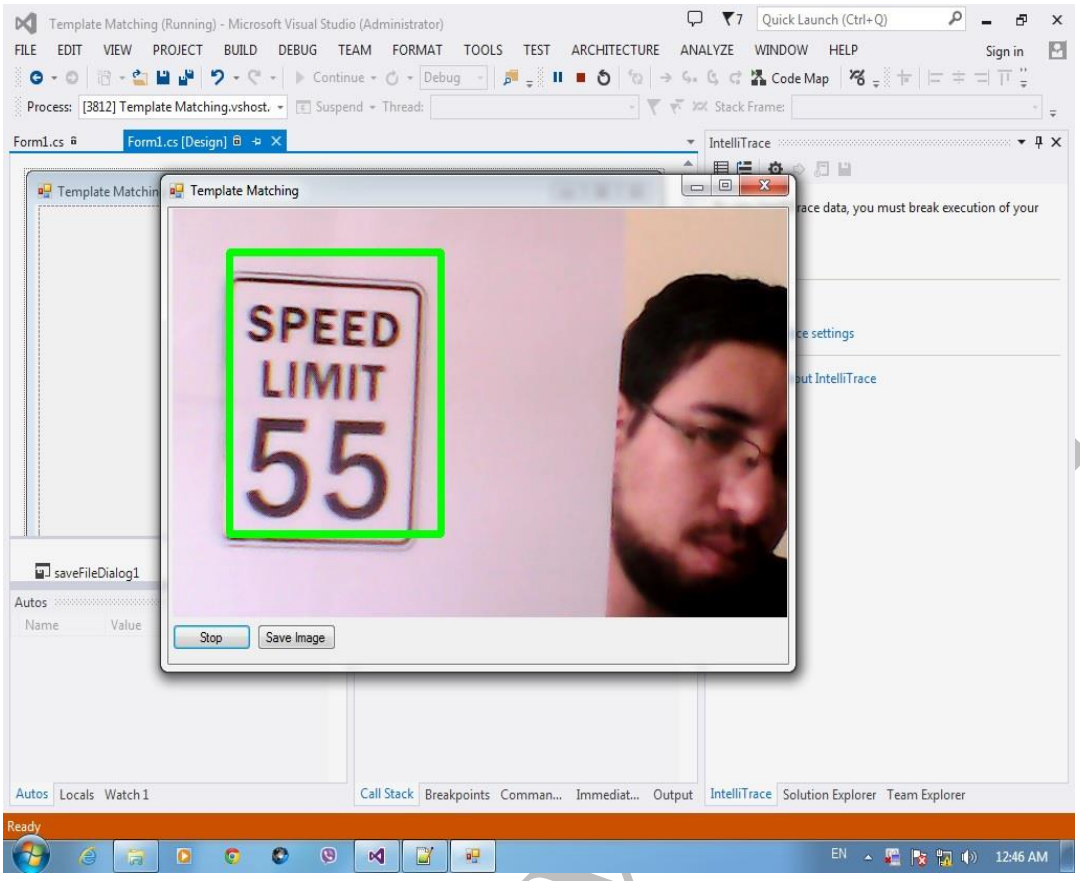
```
if (maxval > thresh)
                {
Cv.Rectangle(img, Cv.Point(minloc.X + tpl.Width, minloc.Y +
tpl.Height),Cv.Point(minloc.X, minloc.Y), CvColor.Green, 5, 0, 0);
```

این شرط به برنامه می گوید که اگر بیشترین مقدار شباهت تطبیق (maxval) بزرگتر از thresh = 0.3 است در این صورت الگوی یافت شده همان الگوی مد نظر ما است و در خط بعد کادر سبز رنگ را دور آن رسم می کند.تابع Cv.Rectangle هفت پارامتر می گیرد که پارامتر اول تصویری است که میخواهیم کادر در آن نمایش داده شود .
پارامتر دوم مختصات x و y که نقطه شروع رسم کادر است و پارامتر بعدی مختصات پایان رسم کادر است.
پارامتر بعدی رنگ کادر را مشخص میکند CvColor.Green و پارامتر بعدی که عدد ۵ می باشد که ضخامت کادر را مشخص میکند و دو پارامتر بعدی که صفر هستند به ترتیب نوع رسم خط و مقدار جابجایی خط رامشخص میکند که مهم نیستد و در صورت تغییر دادن آنها می توان مدل خروجی را مشاهده کرد.
در انتهای حلقه while تصویر پردازش شده را در picture box قرار می دهیم و این کار با دستورات :

```
Bitmap bm = BitmapConverter.ToBitmap(img);
bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height);
pctCvWindow.Image = bm;
```

انجام می شود که مانند مثال فصل قبل بوده و نیازی به توضیح اضافی ندارد.

در انتها کل کد را به صورت یکجا قرار می دهیم تا درصورت دلخواه بیشتر بررسی شود.



شکل (۴-۷) خروجی تطبیق الگو

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using OpenCvSharp;
using System.Threading;

namespace Template_Matching
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }


        //thread
        private Thread _cameraThread;
```

```
        //start thread
        #region Camera Thread
        private void CaptureCamera()
        {
            _cameraThread = new Thread(new ThreadStart(CaptureCameraCallback));
            _cameraThread.Start();
        }

        private void CaptureCameraCallback()
        {
            CvCapture cap = CvCapture.FromCamera(1);
            IplImage tpl = new IplImage("speedlimit55.jpg", LoadMode.AnyColor |
LoadMode.AnyDepth);
            CvPoint minloc;
            CvPoint maxloc;
            double minval;
            double maxval;
            double thresh = 0.3;
            while (true)
                {
                    IplImage img = cap.QueryFrame();
                    IplImage res = Cv.CreateImage(Cv.Size(img.Width - tpl.Width + 1,
img.Height - tpl.Height + 1), BitDepth.F32, 1);
                    IplImage gimg = Cv.CreateImage(Cv.GetSize(img), img.Depth, 1);
                    IplImage gtpl = Cv.CreateImage(Cv.GetSize(tpl), img.Depth, 1);
                    Cv.CvtColor(img, gimg, ColorConversion.BgrToGray);
                    Cv.CvtColor(tpl, gtpl, ColorConversion.BgrToGray);
                    Cv.MatchTemplate(gimg, gtpl, res,
MatchTemplateMethod.CCoeffNormed);
                    Cv.MinMaxLoc(res, out minval, out maxval, out minloc, out maxloc,
null);
                    if (maxval > thresh)
                    {
                        Cv.Rectangle(img, Cv.Point(minloc.X + tpl.Width, minloc.Y +
tpl.Height), Cv.Point(minloc.X, minloc.Y), CvColor.Green, 5, 0, 0);
                    }
                    Bitmap bm = BitmapConverter.ToBitmap(img);
                    bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height);
                    pctCvWindow.Image = bm;
                }

        }
        #endregion

        private void btnStart_Click(object sender, EventArgs e)
        {
            if (btnStart.Text.Equals("Start"))
            {
                CaptureCamera();
                btnStart.Text = "Stop";
            }
            else
            {
                _cameraThread.Abort();
                btnStart.Text = "Start";
            }


        }

        #region Form Handlers
        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
```

```
        {
            if (_cameraThread != null && _cameraThread.IsAlive)
            {
                _cameraThread.Abort();
            }
        }
        #endregion

        private void button1_Click(object sender, EventArgs e)
        {
            saveFileDialog1.FileName = "opencv-image.jpg";
            if (saveFileDialog1.ShowDialog() == DialogResult.OK)
            {
                String imageFileName = saveFileDialog1.FileName;
                pctCvWindow.Image.Save(imageFileName);
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }


    }
}
```

Ehsanavr.com

۷-۳- نتیجه گیری

در این فصل نحوه ی تطبیق الگو در برنامه نویسی را بررسی کردیم و دریافتیم که بیشتر این الگوریتم ها با داده های آماری و علم آمار سروکار دارند.

فـصل ۸ :

واقعیت افزوده با opencv در

Windows Form Application C#

Ehsanavr.com

۸-۱- مقدمه

در این فصل ابتدا توضیح مختصری در مورد واقیعت افزوده می دهیم سپس در قالب مثال ، ابتدایی ترین برنامه ای را که می شود در مورد واقعیت افزوده نوشت را بررسی می کنیم.
لازم به ذکر است که مباحثی چون واقعیت افزوده با توسعه و افزایش سرعت پردازشگر های امروزی قابلیت های بسیاری را به زندگی بشر می دهند و امروزه بیش از پیش اطلاعات و اخبار در مورد این مباحث شنیده می شود.

۸-۲- واقعیت افزوده چیست؟

واقعیت افزوده (به انگلیسی: Augmented Reality) یک نمای فیزیکی زنده ، مستقیم یا غیرمستقیم (ومعمولاً در تعامل با کاربر) است ، که عناصری را پیرامون دنیای واقعی افراد اضافه می کند. این عناصر براساس تولیدات کامپیوتری که از طریق دریافت و پردازش اطلاعات کاربر توسط سنسورهای ورودی مانند صدا، ویدئو، تصاویرگرافیکی یا دادههای GPS می باشد ایجاد می شود. واقعیت رایانه ای مفهوم کلی واقعیت افزوده است. در واقعیت افزوده معمولاً چیزی کم نمی شود بلکه فقط اضافه می شود. همچنین واقعیت افزوده تاحدودی شبیه به واقعیت مجازی است که توسط یک شبیه ساز، دنیای واقعی راکاملا شبیه سازی می کند. در واقع وجه تمایز بین واقعیت مجازی و واقعیت افزوده این است که در واقعیت مجازی کلیه عناصردرک شده توسط کاربر، ساخته شده توسط کامپیوتر هستند. اما در واقعیت افزوده بخشی از اطلاعاتی راکه کاربردرک میکند،دردنیای واقعی وجود دارند و بخشی توسط کامپیوتر ساخته شده اند.

در واقعیت افزوده، عناصرمعمولا به صورت بیدرنگ نگاشته شده و به طور هوشمند مرتبط با عناصرمحیطی می باشند ، مانند نمایش امتیازمسابقات ورزشی در زمان پخش از تلویزیون. با کمک تکنولوژی پیشرفته واقعیت افزوده (برای مثال افزودن قابلیت بینایی کامپیوتری و تشخیص اشیاء) میتوان اطلاعات مرتبط با دنیای واقعی پیرامون کاربررا به صورت تعاملی ودیجیتالی به او ارائه کرد. همچنین می توان اطلاعات مرتبط با محیط و اشیاء اطراف را برروی دنیای واقعی نگاشت. ایده اولیه واقعیت افزوده اولین باردر سال ۱۹۹۰ توسط توماس کادل کارمند بوئینگ مطرح شد.

Ehsanavr.com

شکل (۸-۱) واقعیت افزوده

تصویر بالا نمونه طرحی است که توسط نویسنده در قالب اپلیکیشنی برای گوشی های اندروید نوشته شده است.
این برنامه توسط گیم انجین یونیتی و با زبان برنامه نویسی سی شارپ نوشته شده است.

۸-۲-۲- کاربرد ها

- ورزش

ساده ترین نمونه واقعیت مجازی را می توان در برنامه های ورزشی تلویزیونی مشاهده نمود. برای مثال، نمایش نتیجه مسابقات فوتبال دردایره مرکز زمین، یا در زمان پخش مسابقات شنا در تلویزیون، نمایش شماره خط هر شناگر و نمایش رکورد در پشت هر شناگر(که امکان مقایسه شناگران را به بینندگان مسابقه می دهد) نمونه هایی عادی ازکاربرد این سیستم است.

- بازی های ویدئویی دستی

بوسیله "کارت واقعیت افزوده" برای نینتندو۳ دی اس و پلی استیشن ویتا با استفاده ازدوربین مخصوص، میتوان بازی کرد.

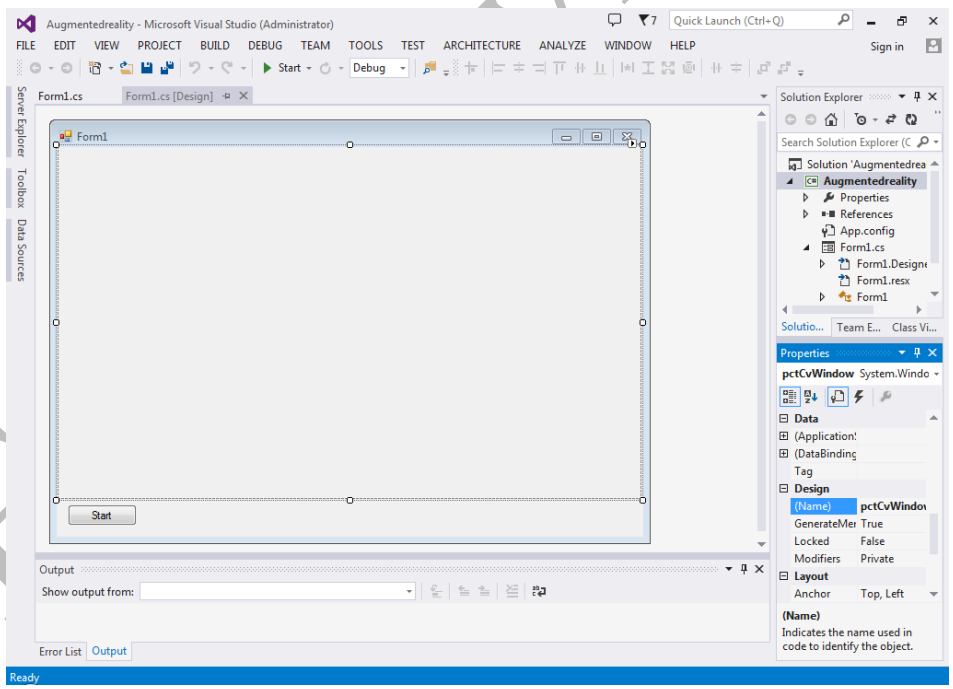
Ehsanavr.com

- نرم افزار های کاربردی

با توجه به گسترش سریع موبایل های هوشمند (Smart Phone) به عنوان اصلی ترین سخت افزارها برای " واقعیت افزوده "می توان به گسترش روز افزون نرم افزارهای کاربردی در حوزه های مختلف اشاره کرد.

۸-۳- ایجاد یک نمونه ساده از پیاده سازی واقعیت افزوده

ابتدا در visual studio یک پروژه از نوع Windows Form Application ایجاد می کنیم و زبان برنامه نویسی مد نظر ما در این مثال سی شارپ خواهد بود.
در طراحی فرم از یک button با مشخصه نام btnstart و نام روی دکمه Start استفاده می کنیم و یک عدد picture box نیز با نام pctCVwindow به فرم اضافه می کنیم .



شکل (۸-۲) ایجاد پروژه در ویژوال استودیو

محیط کد زنی نیز مانند مثال های فصل قبل است و ما از یک thread جهت فراخوانی دوربین استفاده می کنیم و علت استفاده از thread هم این بود که مانع از crash کردن ناگهانی برنامه شویم .

بر روی دکمه Start دو بار کلیک کنید و وارد تابع این دکمه در محیط برنامه نویسی شوید و کد زیر را بنویسید.

```
private void btnStart_Click(object sender, EventArgs e)
{
    if (btnStart.Text.Equals("Start"))
    {
        CaptureCamera();
        btnStart.Text = "Stop";
    }
    else
    {
        Cv.ReleaseCapture(cap);
        _cameraThread.Abort();
        btnStart.Text = "Start";
        pctCvWindow.Image = null;
    }
}
```

این کد مانند مثال های فصل قبل است ولی اندکی آن را تغییر داده ایم این تغییرات در بخش مربوط به کدهای زیر است.

```
Cv.ReleaseCapture(cap);
pctCvWindow.Image = null;
```

خط اول این تغییرات مربوط به آزاد سازی سخت افزار گرفته شده (دوربین) جهت استفاده های بعدی می باشد و خط دوم باعث میشود که وقتی کاربر بر روی دکمه Stop کلیک کرد picture box مورد نظر به حالت اولیه خود یعنی حالت شروع برنامه باز گردد.

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (_cameraThread != null && _cameraThread.IsAlive)
    {
        _cameraThread.Abort();
    }
}
```

کد بالا مربوط به یک event برای فرم است که هنگام بسته شدن برنامه thread گرفته شده را آزاد می کند.

```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    Cv.ReleaseCapture(cap);
    _cameraThread.Abort();

}
```

این نیز یک event است که در صورت بسته شدن ناگهانی برنامه و یا crash کردن منابع مربوط به سخت افزار دوربین را آزاد می کند تا جهت استفاده های بعدی در دسترس باشد.

Ehsanavr.com

در داخل کلاس Form خود ، تعریف مقادیر متغیر های مورد استفاده در برنامه را داریم که باعث میشود در تمام برنامه به صورت سراسری به آنها دسترسی داشته باشیم.
این متغیر ها در زیر آمده اند

```
//define variables
public float alpha = 0.1f;
public float beta = 1.0f;
CvCapture cap;
IplImage tpl;
CvPoint minloc;
CvPoint maxloc;
double minval;
double maxval;
IplImage img;
IplImage res;
IplImage flagimage;
IplImage flagresized;
public double thresh = 0.3;
//create thread
private Thread _cameraThread;
//camera & etc
```

سپس تابع مربوط به فراخوانی thread و استارت تابع پردازش تصویر خود را در قالب یک thread مستقل داریم .

```
        private void CaptureCamera()
        {
            _cameraThread = new Thread(new ThreadStart(CaptureCameraCallback));
            _cameraThread.Start();
        }
```

این کد تابع اصلی ما را که پردازش تصویر در آن انجام میشود یعنی () CaptureCameraCallback را فراخوانی می کند و مانند فصل های قبل بدون تغییر است.

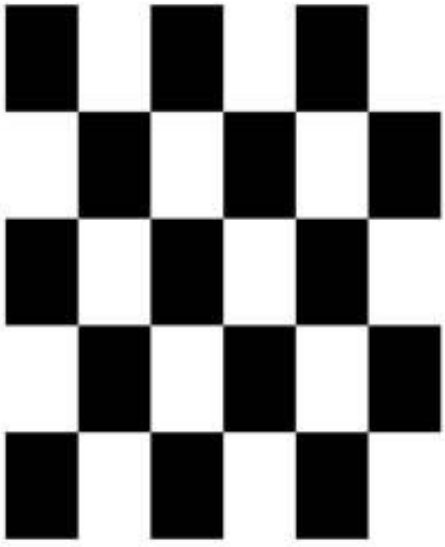
سپس می رسیم به توضیحات مربوط به تابع CaptureCameraCallback که تمامی عملیات مربوط به پردازش تصویر در آن انجام می شود .

```
        private void CaptureCameraCallback()
        {
            CvCapture cap = CvCapture.FromCamera(1);
            tpl = Cv.LoadImage("image.jpg", LoadMode.Color);
            flagimage = Cv.LoadImage("mamad.jpg", LoadMode.AnyColor);
                while (true)
                {
                    img = cap.QueryFrame();
                    res = Cv.CreateImage(Cv.Size(img.Width - tpl.Width + 1, img.Height
- tpl.Height + 1), BitDepth.F32, 1);
                    IplImage gimg = Cv.CreateImage(Cv.GetSize(img), img.Depth, 1);
                    IplImage gtpl = Cv.CreateImage(Cv.GetSize(tpl), img.Depth, 1);
                    Cv.CvtColor(img, gimg, ColorConversion.BgrToGray);
                    Cv.CvtColor(tpl, gtpl, ColorConversion.BgrToGray);
                    Cv.MatchTemplate(gimg, gtpl, res,
MatchTemplateMethod.CCoeffNormed);
                    Cv.MinMaxLoc(res, out minval, out maxval, out minloc, out maxloc,
null);
```

ehsanavr.com

```
                    CvRect cropRect = new CvRect(minloc.X, minloc.Y, tpl.Width,
tpl.Height);
                    flagresized = new IplImage(tpl.Size, flagimage.Depth,
flagimage.NChannels);
                    flagimage.Resize(flagresized);
                    if (maxval > thresh)
                    {
                        Cv.Rectangle(img, Cv.Point(minloc.X + tpl.Width, minloc.Y +
tpl.Height), Cv.Point(minloc.X, minloc.Y), CvColor.Green, 5, 0, 0);
                        Cv.SetImageROI(img, cropRect);
                        Cv.AddWeighted(img, alpha, flagresized, beta, 0.0f, img);
                        Cv.ResetImageROI(img);
                    }
                    Bitmap bm = BitmapConverter.ToBitmap(img);
                    bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height);
                    pctCvWindow.Image = bm;
                }
            }
```

ابتدا نیاز است که نحوه کار برنامه اندکی توضیح داده شود تا شاید فهمیدن روش کار کد ها آسان تر گردد. این برنامه در ابتدا ورودی را از دوربین میگیرد وبه دنبال یک الگوی از قبل معرفی شده به برنامه در تصاویر ورودی توسط دوربین میگردد. این الگو در پوشه bin/debug قرار داده شد و نام آن image.jpg است و شکل آن به صورت زیر است.



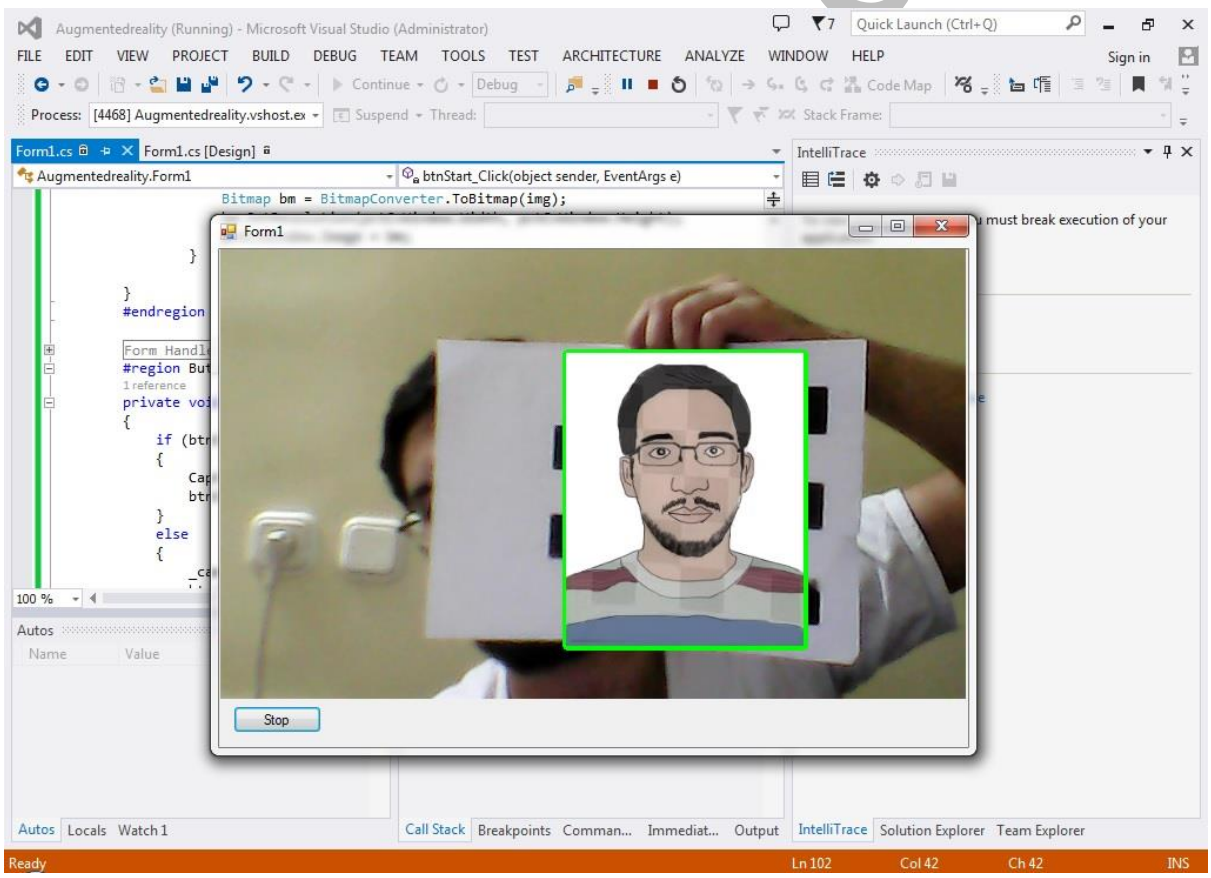
شکل (۸-۳) الگوی معرفی شده به برنامه

این یک صفحه شطرنجی است. برنامه در صورت یافتن این الگو در تصاویر ورودی دوربین اقدام به پروجکت کردن و انداختن یک تصویر دیگر که آن نیز از قبل به برنامه داده شده می کند و آن تصویر مربوط به شخص نویسنده است که با فتوشاپ طراحی شده است.

Ehsanvr.com

شکل (۸-۴) تصویر نویسنده

و در نهایت حاصل کلی این برنامه به صورت عکس زیر در می آید.



شکل (۸-۵) خروجی برنامه

اما توضیحات مربوط به کد .

در داخل تابع CaptureCameraCallback سه خط اول برنامه

```
CvCapture cap = CvCapture.FromCamera(1);
tpl = Cv.LoadImage("image.jpg", LoadMode.Color);
flagimage = Cv.LoadImage("mamad.jpg", LoadMode.AnyColor);
```

به ترتیب ورودی دوربین را می گیرد و سپس در متغیر tpl صفحه شطرنجی را بارگذاری می کند و در متغیر flagimage عکس شخص نویسنده را بارگذاری می کند.

سپس درون حلقه while عملیات تطبیق الگو انجام می شود که تا حدود زیادی مانند فصل قبل است و توضیحات آن به طور مفصل در فصول قبل آمده است. ما فقط اقدام به توضیح کد های جدید و متفاوت از فصول قبل می کنیم تا از زیاده گویی بپرهیزیم.

خط مربوط به :

```
CvRect cropRect = new CvRect(minloc.X, minloc.Y, tpl.Width, tpl.Height);
```

پس از یافتن الگو در تصویر دوربین اقدام به ایجاد یک چهارچوب دور الگوی یافت شده می کند.

```
flagresized = new IplImage(tpl.Size, flagimage.Depth, flagimage.NChannels);
flagimage.Resize(flagresized);
```

دو خط بالا تصویر لود شده نویسنده را متناسب با اندازه صفحه شطرنجی تغییر اندازه می دهد. علت این امر این است که در خطوط بعدی که میخواهیم تصویر نویسنده را بر روی صفحه بیاندازیم با خطای متفاوت بودن اندازه ها مواجه نشویم.

کد های مربوط به بررسی شرط if مهمترین کد در داخل حلقه while است.

```
if (maxval > thresh)
                    {
                        Cv.Rectangle(img, Cv.Point(minloc.X + tpl.Width, minloc.Y +
tpl.Height), Cv.Point(minloc.X, minloc.Y), CvColor.Green, 5, 0, 0);
                        Cv.SetImageROI(img, cropRect);
                        Cv.AddWeighted(img, alpha, flagresized, beta, 0.0f, img);
                        Cv.ResetImageROI(img);
                    }
```

مانند فصل قبل این کد ها کار مشابهی انجام می دهند وفقط سه خط آخر مربوط به شرط ، وظیفه گذاشتن عکس نویسنده روی الگوی یافت شده را ایفا می کند.

کد مربوط به Cv.SetImageROI() آن قسمت از تصویر ورودی دوربین که الگو در آن یافت شده را برش میدهد .

RoI مخفف عبارت Region Of Interest است که ناحیه مورد نظر از تصویر را انتخاب و برش می دهد . پارامتر های این تابع به ترتیب تصویر ورودی از دوربین و مختصات چهارچوبی است که می خواهیم تصویر را انتخاب و برش دهیم.این چهارچوب همان الگوی یافت شده در تصویر است.

در خط بعد در قسمت مربوط به کد :

```
Cv.AddWeighted(img, alpha, flagresized, beta, 0.0f, img);
```

تصویر را به روی صفحه شطرنجی می اندازیم.

پارامتر اول این تابع ورودی تصویر است که می خواهیم تصویر روی آن قرار گیرد .پارامتر دوم alpha میزان شفافیت تصویر انداخته شده (تصویر نویسنده) را مشخص میکند. پارامتر بعدی یعنی flagresized همان تصویر نویسنده است که تغییر اندازه داده شده است .

دو پارامتر بعدی مربوط به شدت و ویژگی های تصویر است که beta و gamma = 0 است.

در نهایت آخرین پارامتر ، خروجی است که ما برابر با همان ورودی یعنی (img) قرارش دادیم که به صورت فلیتری بر روی ورودی عمل کند تا نیازی به تعریف متغیر جدید نباشد.

خط مربوط به :

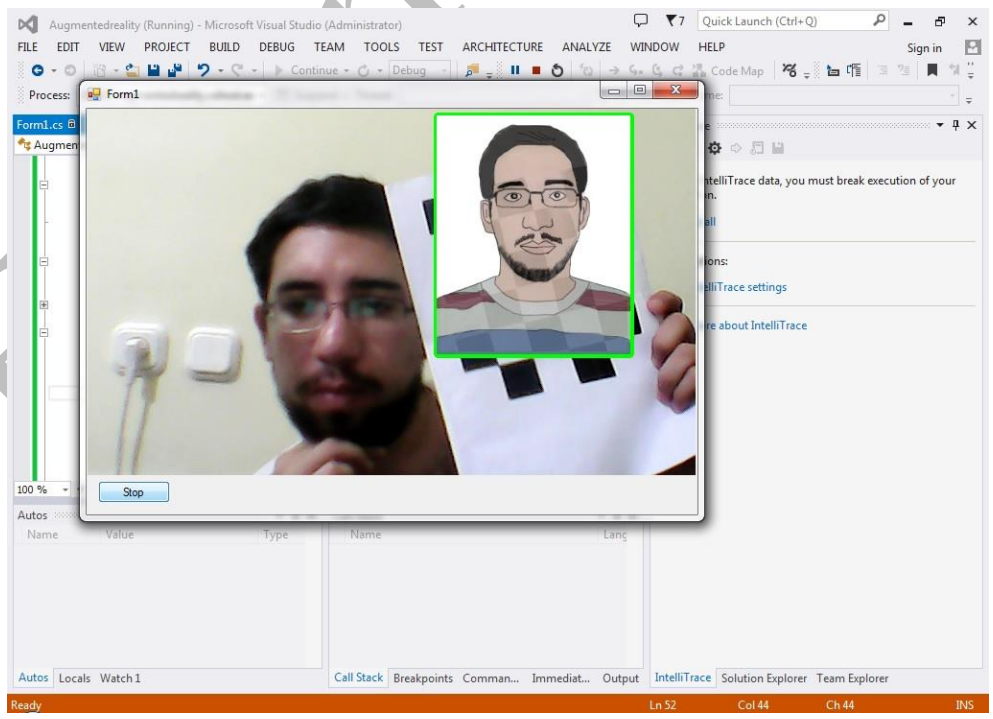
```
Cv.ResetImageROI(img);
```

ناحیه انتخاب شده و برش داده شده را آزاد میکند .

توجه:

در صورت استفاده از تابع Cv.SetImageROI پس از انجام تغییرات دلخواه حتما باید در انتها تابع Cv.ResetImageROI فراخوانی کنیم در غیر اینصورت برنامه با خطا مواجه خواهد شد.

در نهایت در بیرون از حلقه تصویر را ابتدا به استاندارد Bitmap تبدیل کرده و در picture box قرار میدهیم.



شکل (۸-۶) خروجی نهایی برنامه

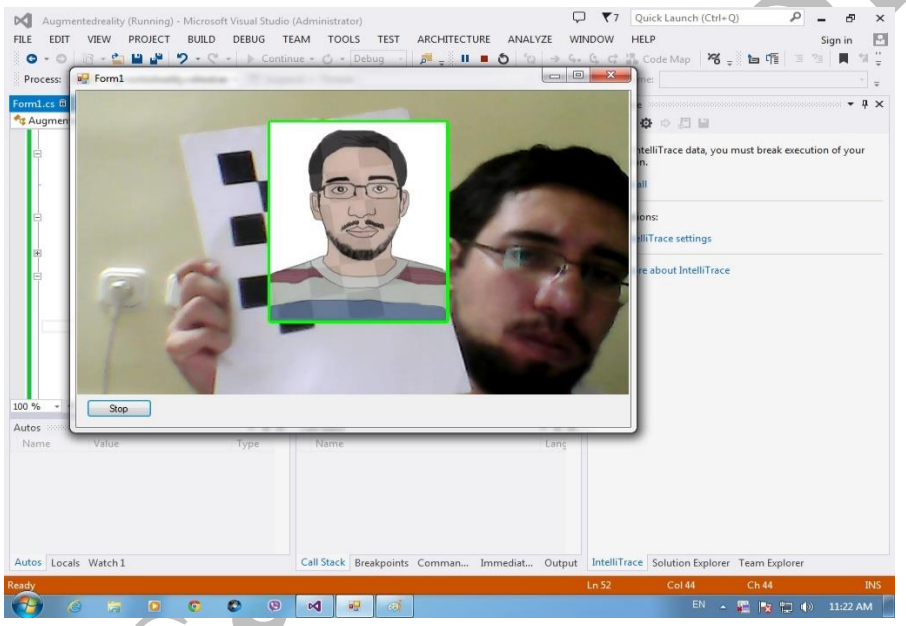
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using OpenCvSharp;
using System.Threading;

namespace Augmentedreality
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      //define variables
      public float alpha = 0.1f;
      public float beta = 1.0f;
      CvCapture cap;
      IplImage tpl;
      CvPoint minloc;
      CvPoint maxloc;
      double minval;
      double maxval;
      IplImage img;
      IplImage res;
      IplImage flagimage;
      IplImage flagresized;
      public double thresh = 0.3;
      //create thread
      private Thread _cameraThread;
      //camera & etc
      #region Camera Thread
     //start thread
      private void CaptureCamera()
      {
          _cameraThread = new Thread(new ThreadStart(CaptureCameraCallback));
         _cameraThread.Start();
      }
      private void CaptureCameraCallback()
      {
         CvCapture cap = CvCapture.FromCamera(1);
         tpl = Cv.LoadImage("image.jpg", LoadMode.Color);
         flagimage = Cv.LoadImage("mamad.jpg", LoadMode.AnyColor);
```

sanavr.com

```
            while (true)
            {
                img = cap.QueryFrame();
                res = Cv.CreateImage(Cv.Size(img.Width - tpl.Width + 1, img.Height -
tpl.Height + 1), BitDepth.F32, 1);
                IplImage gimg = Cv.CreateImage(Cv.GetSize(img), img.Depth, 1);
                IplImage gtpl = Cv.CreateImage(Cv.GetSize(tpl), img.Depth, 1);
                Cv.CvtColor(img, gimg, ColorConversion.BgrToGray);
                Cv.CvtColor(tpl, gtpl, ColorConversion.BgrToGray);
                Cv.MatchTemplate(gimg, gtpl, res, MatchTemplateMethod.CCoeffNormed);
                Cv.MinMaxLoc(res, out minval, out maxval, out minloc, out maxloc, null);
                CvRect cropRect = new CvRect(minloc.X, minloc.Y, tpl.Width, tpl.Height);
                flagresized = new IplImage(tpl.Size, flagimage.Depth, flagimage.NChannels);
                flagimage.Resize(flagresized);
                if (maxval > thresh)
                {
                    Cv.Rectangle(img, Cv.Point(minloc.X + tpl.Width, minloc.Y + tpl.Height),
Cv.Point(minloc.X, minloc.Y), CvColor.Green, 5, 0, 0);
                    Cv.SetImageROI(img, cropRect);
                    Cv.AddWeighted(img, alpha, flagresized, beta, 0.0f, img);
                    Cv.ResetImageROI(img);
                }
                Bitmap bm = BitmapConverter.ToBitmap(img);
                bm.SetResolution(pctCvWindow.Width, pctCvWindow.Height);
                pctCvWindow.Image = bm;
            }
        }
        #endregion

        #region Form Handlers
        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (_cameraThread != null && _cameraThread.IsAlive)
            {
                _cameraThread.Abort();
            }
        }
        #endregion
        #region Button Handlers
        private void btnStart_Click(object sender, EventArgs e)
        {
            if (btnStart.Text.Equals("Start"))
            {
                CaptureCamera();
                btnStart.Text = "Stop";
            }
            else
            {
                Cv.ReleaseCapture(cap);
                _cameraThread.Abort();
```

arianavr.com

```
            btnStart.Text = "Start";
            pctCvWindow.Image = null;
        }
    }
    #endregion

    private void Form1_FormClosed(object sender, FormClosedEventArgs e)
    {
        Cv.ReleaseCapture(cap);
        _cameraThread.Abort();

    }
  }
}
```



شکل (۸-۷) خروجی نهایی برنامه

۸-۴- نتیجه گیری

در این فصل تمامی مراحل مربوط به برنامه نویسی ساده در رابطه با واقعیت افزوده را گام به گام توضیح دادیم و خروجی حاصل از برنامه را مشاهده کردیم.

Ehsan.com

فـصل ۹ :

جمع‌بندی و پیشنهادها

Ehsanavr.com

۹-۱- مقدمه

همانطور که در فصول قبلی بررسی شد ، ما به معرفی کتابخانه opencv پرداختیم و دریافتیم که این کتابخانه به رایگان در اختیار افراد و شرکت ها قرار گرفته است و تمامی افراد می توانند در توسعه و پیشرفت این مجموعه پردازش تصویر سهیم باشند.
در فصل سوم از این تحقیق ما اقدام به معرفی این کتابخانه پردازش تصویر کردیم و خاستگاه آن و اینکه توسط چه شرکتی توسعه و سرمایه گذاری شده است پرداختیم و همچنین پلتفرم های قابل پشتیبانی را معرفی کردیم و معماری های قابل پشتیبانی را نیز توضیح دادیم .
در فصل چهارم از این پروژه ما نحوه نصب و متصل کردن این کتابخانه به ویژوال استودیو را در قالب تصویر یک به یک شرح دادیم و سپس ساده ترین برنامه موجود که نمایش یک تصویر است را با زبان C++ شرح دادیم.
درفصل پنجم ساختار ها و توابع داخلی این کتابخانه را تا حدودی توضیح دادیم تا بتوان درک کاملی از برنامه نویسی با این مجموعه کتابخانه داشت.
در فصول ششم تا هشتم مثال هایی عملی در قالب زبان برنامه نویسی C# را به این مجموعه اضافه کرده و خط به خط از کد های استفاده شده را توضیح دادیم .
فصل هشتم را می توان چکیده و تکمیل شده ی فصول ششم و هفتم دانست و این فصل ساده ترین مثال کاربردی در رابطه با واقعیت افزوده را بررسی میکند.
امید است که کلیه این مطالب برای شما خوانندگان گرامی ، کاربردی و سرگرم کننده بوده باشد .

۹-۲- محتوا

۹-۲-۱- جمع‌بندی

این تحقیق ایده اولیه پردازش تصاویر و بررسی واقعیت افزوده در پردازش تصویر را در قالب مثال هایی کاربردی برای اشخاص عملی می سازد و لازم به ذکر است که این مثال ها در قالب برنامه هایی جهت روشن شدن دید افراد از اینکه ،کلا پردازش تصاویر دیجیتال در کامپیوتر به چه صورت انجام می شود مورد استفاده قرار میگیرد.

Ehsanavr.com

۹-۲-۲- نوآوری

از جمله نو آوری هایی که در این زمینه opencv انجام گرفته است ، پورت کردن این کتابخانه با زبان های سطح بالا همچون C# بوده که برنامه نویسی را بسیار تسهیل کرده است.

۹-۲-۳- پیشنهادها

در رابطه با این مقاله لازم به ذکر است که افراد بسته به نوع نیاز و کاربردی که برای برنامه نویسی خود در زمینه پردازش تصویر در نظر میگیرند یکی از روش های مطرح شده در این مقاله را برگزینند.
چنانچه در برنامه خود نیاز به سرعت در پردازش تصاویر دارید بهترین گزینه برنامه نویسی با opencv در زبان های C و C++ است.
در صورتی که پیچیدگی بیشتر و سهولت در برنامه نویسی مطرح است و می خواهید برنامه شما کاربردی باشد و نیاز به پردازش سریع تر ندارید ، استفاده از opencvsharp بهترین گزینه پیش رو خواهد بود.
لازم به ذکر است که این تفاوت سرعت معمولا بین چند میلی ثانیه تا چند ده میلی ثانیه می باشد و در برنامه های کاربردی (Applications) قابل اغماض خواهد بود.

Ehsanavr.com

مراجع

Ehsanavr.com

مراجع

[1] Bradski , Gray r ,"Learning opencv : computer vision with the opencv library" , 2008

[2] Robert Laganiere "Opencv 2 computer vision Application programing cookbook" , 2011

[3] M.Agrawal "Opencv reference manual v2.2" ,2010

[4] Shervin Emami , Daniel Lelis baggio "Mastering opencv with practical computer vision projects "

[5] Http://Wikipedia.com "augmented reality " 2014

[6] Jayneil Dalal ,Sohil Patel "Beginner's Guide to Computer vision" , 2013

[7] Romixery@pc7 "learning opencv" , 2013

[8] **Http://code.google.com/p/Opencvsharp**

[9] **Http://Doc.opencv.org**

[10] **http://www.youtube.com/user/khounslow**

[11] **Http://www.prodigyproductionsllc.com**

[12] **Http://Emgucv.com/wiki**

[13] **Http://parvizsaiyadi.com/**

[۱۴] الهام شعبانی نیا – رسول محمد نصیری "بینایی ماشین و پردازش تصویر با opencv" انتشارات دانشگاهی کیان ۱۳۹۲

[۱۵] مهندس حسن جعفری "نحوه افزودن کتابخانه opencv به ویژوال استودیو "

[۱۶] کیام نصیری "آموزش نصب opencv و افزودن آن به ویژوال استودیو" انتشاران دانشگاه کردستان ۲۰۰۸

Ehsanavr.com

پیوست‌ها

Ehsanavr.com

OpenCV 2.4 Cheat Sheet (C++)

*The OpenCV C++ reference manual is here: http://docs.opencv.org. Use **Quick Search** to find descriptions of the particular functions and classes*

Key OpenCV Classes

Point_	Template 2D point class
Point3_	Template 3D point class
Size_	Template size (width, height) class
Vec	Template short vector class
Matx	Template small matrix class
Scalar	4-element vector
Rect	Rectangle
Range	Integer value range
Mat	2D or multi-dimensional dense array (can be used to store matrices, images, histograms, feature descriptors, voxel volumes etc.)
SparseMat	Multi-dimensional sparse array
Ptr	Template smart pointer class

Matrix Basics

Create a matrix

```
Mat image(240, 320, CV_8UC3);
```

[Re]allocate a pre-declared matrix

```
image.create(480, 640, CV_8UC3);
```

Create a matrix initialized with a constant

```
Mat A33(3, 3, CV_32F, Scalar(5));
Mat B33(3, 3, CV_32F); B33 = Scalar(5);
Mat C33 = Mat::ones(3, 3, CV_32F)*5.;
Mat D33 = Mat::zeros(3, 3, CV_32F) + 5.;
```

Create a matrix initialized with specified values

```
double a = CV_PI/3;
Mat A22 = (Mat_<float>(2, 2) <<
  cos(a), -sin(a), sin(a), cos(a));
float B22data[] = {cos(a), -sin(a), sin(a), cos(a)};
Mat B22 = Mat(2, 2, CV_32F, B22data).clone();
```

Initialize a random matrix

```
randu(image, Scalar(0), Scalar(256)); // uniform dist
randn(image, Scalar(128), Scalar(10)); // Gaussian dist
```

Convert matrix to/from other structures (without copying the data)

```
Mat image_alias = image;
float* Idata=new float[480*640*3];
Mat I(480, 640, CV_32FC3, Idata);
vector<Point> iptvec(10);
Mat iP(iptvec); // iP - 10x1 CV_32SC2 matrix
IplImage* oldC0 = cvCreateImage(cvSize(320,240),16,1);
Mat newC = cvarrToMat(oldC0);
IplImage oldC1 = newC; CvMat oldC2 = newC;
```

... (with copying the data)

```
Mat newC2 = cvarrToMat(oldC0).clone();
vector<Point2f> ptvec = Mat_<Point2f>(iP);
```

Access matrix elements

```
A33.at<float>(i,j) = A33.at<float>(j,i)+1;
Mat dyImage(image.size(), image.type());
for(int y = 1; y < image.rows-1; y++) {
  Vec3b* prevRow = image.ptr<Vec3b>(y-1);
  Vec3b* nextRow = image.ptr<Vec3b>(y+1);
  for(int x = 0; x < image.cols; x++)
    for(int c = 0; c < 3; c++)
     dyImage.at<Vec3b>(y,x)[c] =
     saturate_cast<uchar>(
     nextRow[x][c] - prevRow[x][c]);
}
Mat_<Vec3b>::iterator it = image.begin<Vec3b>(),
  itEnd = image.end<Vec3b>();
for(; it != itEnd; ++it)
  (*it)[1] ^= 255;
```

Matrix Manipulations: Copying, Shuffling, Part Access

src.copyTo(dst)	Copy matrix to another one
src.convertTo(dst,type,scale,shift)	Scale and convert to another datatype
m.clone()	Make deep copy of a matrix
m.reshape(nch,nrows)	Change matrix dimensions and/or number of channels without copying data
m.row(i), m.col(i)	Take a matrix row/column
m.rowRange(Range(i1,i2)) m.colRange(Range(j1,j2))	Take a matrix row/column span
m.diag(i)	Take a matrix diagonal
m(Range(i1,i2),Range(j1,j2)), m(roi)	Take a submatrix
m.repeat(ny,nx)	Make a bigger matrix from a smaller one
flip(src,dst,dir)	Reverse the order of matrix rows and/or columns
split(...)	Split multi-channel matrix into separate channels
merge(...)	Make a multi-channel matrix out of the separate channels
mixChannels(...)	Generalized form of split() and merge()
randShuffle(...)	Randomly shuffle matrix elements

Example 1. Smooth image ROI in-place

```
Mat imgroi = image(Rect(10, 20, 100, 100));
GaussianBlur(imgroi, imgroi, Size(5, 5), 1.2, 1.2);
```

Example 2. Somewhere in a linear algebra algorithm

```
m.row(i) += m.row(j)*alpha;
```

Example 3. Copy image ROI to another image with conversion

```
Rect r(1, 1, 10, 20);
Mat dstroi = dst(Rect(0,10,r.width,r.height));
src(r).convertTo(dstroi, dstroi.type(), 1, 0);
```

Simple Matrix Operations

OpenCV implements most common arithmetical, logical and other matrix operations, such as

- add(), subtract(), multiply(), divide(), absdiff(), bitwise_and(), bitwise_or(), bitwise_xor(), max(), min(), compare()
  – correspondingly, addition, subtraction, element-wise multiplication ... comparison of two matrices or a matrix and a scalar.

  Example. Alpha compositing function:

```
void alphaCompose(const Mat& rgba1,
    const Mat& rgba2, Mat& rgba_dest)
{
    Mat a1(rgba1.size(), rgba1.type()), ra1;
    Mat a2(rgba2.size(), rgba2.type());
    int mixch[]={3, 0, 3, 1, 3, 2, 3, 3};
    mixChannels(&rgba1, 1, &a1, 1, mixch, 4);
    mixChannels(&rgba2, 1, &a2, 1, mixch, 4);
    subtract(Scalar::all(255), a1, ra1);
    bitwise_or(a1, Scalar(0,0,0,255), a1);
    bitwise_or(a2, Scalar(0,0,0,255), a2);
    multiply(a2, ra1, a2, 1./255);
    multiply(a1, rgba1, a1, 1./255);
    multiply(a2, rgba2, a2, 1./255);
    add(a1, a2, rgba_dest);
}
```

- sum(), mean(), meanStdDev(), norm(), countNonZero(), minMaxLoc(),
  – various statistics of matrix elements.
- exp(), log(), pow(), sqrt(), cartToPolar(), polarToCart()
  – the classical math functions.
- scaleAdd(), transpose(), gemm(), invert(), solve(), determinant(), trace(), eigen(), SVD,
  – the algebraic functions + SVD class.
- dft(), idft(), dct(), idct(),
  – discrete Fourier and cosine transformations

For some operations a more convenient algebraic notation can be used, for example:

```
Mat delta = (J.t()*J + lambda*
   Mat::eye(J.cols, J.cols, J.type()))
   .inv(CV_SVD)*(J.t()*err);
```

implements the core of Levenberg-Marquardt optimization algorithm.

Image Processsing

Filtering

filter2D()	Non-separable linear filter
sepFilter2D()	Separable linear filter
boxFilter(), GaussianBlur(), medianBlur(), bilateralFilter()	Smooth the image with one of the linear or non-linear filters
Sobel(), Scharr()	Compute the spatial image derivatives
Laplacian()	compute Laplacian: $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$
erode(), dilate()	Morphological operations

Example. Filter image in-place with a 3x3 high-pass kernel (preserve negative responses by shifting the result by 128):

```
filter2D(image, image, image.depth(), (Mat_<float>(3,3)<
    -1, -1, -1, -1, 9, -1, -1, -1, -1), Point(1,1), 128);
```

Geometrical Transformations

resize()	Resize image
getRectSubPix()	Extract an image patch
warpAffine()	Warp image affinely
warpPerspective()	Warp image perspectively
remap()	Generic image warping
convertMaps()	Optimize maps for a faster remap() execution

Example. Decimate image by factor of $\sqrt{2}$:

```
Mat dst; resize(src, dst, Size(), 1./sqrt(2), 1./sqrt(2));
```

Various Image Transformations

cvtColor()	Convert image from one color space to another
threshold(), adaptivethreshold()	Convert grayscale image to binary image using a fixed or a variable threshold
floodFill()	Find a connected component using region growing algorithm
integral()	Compute integral image
distanceTransform()	build distance map or discrete Voronoi diagram for a binary image.
watershed(), grabCut()	marker-based image segmentation algorithms. See the samples watershed.cpp and grabcut.cpp.

Histograms

calcHist()	Compute image(s) histogram
calcBackProject()	Back-project the histogram
equalizeHist()	Normalize image brightness and contrast
compareHist()	Compare two histograms

Example. Compute Hue-Saturation histogram of an image:

```
Mat hsv, H;
cvtColor(image, hsv, CV_BGR2HSV);
int planes[]={0, 1}, hsize[] = {32, 32};
calcHist(&hsv, 1, planes, Mat(), H, 2, hsize, 0);
```

Contours

See contours2.cpp and squares.cpp samples on what are the contours and how to use them.

Data I/O

XML/YAML storages are collections (possibly nested) of scalar values, structures and heterogeneous lists.

Writing data to YAML (or XML)

```
// Type of the file is determined from the extension
FileStorage fs("test.yml", FileStorage::WRITE);
fs « "i" « 5 « "r" « 3.1 « "str" « "ABCDEFGH";
fs « "mtx" « Mat::eye(3,3,CV_32F);
fs « "mylist" « "[" « CV_PI « "1+1" «
    "{:" « "month" « 12 « "day" « 31 « "year"
    « 1969 « "}" « "]";
fs « "mystruct" « "{" « "x" « 1 « "y" « 2 «
    "width" « 100 « "height" « 200 « "lbp" « "[:";
const uchar arr[] = {0, 1, 1, 0, 1, 1, 0, 1};
fs.writeRaw("u", arr, (int)(sizeof(arr)/sizeof(arr[0])));
fs « "]" « "}";
```

Scalars (integers, floating-point numbers, text strings), matrices, STL vectors of scalars and some other types can be written to the file storages using « operator

Reading the data back

```
// Type of the file is determined from the content
FileStorage fs("test.yml", FileStorage::READ);
int i1 = (int)fs["i"]; double r1 = (double)fs["r"];
string str1 = (string)fs["str"];
Mat M; fs["mtx"] » M;
FileNode tl = fs["mylist"];
CV_Assert(tl.type() == FileNode::SEQ && tl.size() == 3);
double tl0 = (double)tl[0]; string tl1 = (string)tl[1];
int m = (int)tl[2]["month"], d = (int)tl[2]["day"];
int year = (int)tl[2]["year"];
FileNode tm = fs["mystruct"];
Rect r; r.x = (int)tm["x"], r.y = (int)tm["y"];
r.width = (int)tm["width"], r.height = (int)tm["height"];
int lbp_val = 0;
FileNodeIterator it = tm["lbp"].begin();
for(int k = 0; k < 8; k++, ++it)
   lbp_val |= ((int)*it) « k;
```

Scalars are read using the corresponding FileNode's cast operators. Matrices and some other types are read using » operator. Lists can be read using FileNodeIterator's.

Writing and reading raster images

```
imwrite("myimage.jpg", image);
Mat image_color_copy = imread("myimage.jpg", 1);
Mat image_grayscale_copy = imread("myimage.jpg", 0);
```

The functions can read/write images in the following formats: ***BMP (.bmp), JPEG (.jpg, .jpeg), TIFF (.tif, .tiff), PNG (.png), PBM/PGM/PPM (.p?m), Sun Raster (.sr), JPEG 2000 (.jp2).*** *Every format supports 8-bit, 1- or 3-channel images. Some formats (PNG, JPEG 2000) support 16 bits per channel.*

Reading video from a file or from a camera

```
VideoCapture cap;
if(argc > 1) cap.open(string(argv[1])); else cap.open(0);
Mat frame; namedWindow("video", 1);
for(;;) {
   cap » frame; if(!frame.data) break;
   imshow("video", frame); if(waitKey(30) >= 0) break;
}
```

Simple GUI (highgui module)

namedWindow(winname,flags)	Create named highgui window
destroyWindow(winname)	Destroy the specified window
imshow(winname, mtx)	Show image in the window
waitKey(delay)	Wait for a key press during the specified time interval (or forever). Process events while waiting. *Do not forget to call this function several times a second in your code.*
createTrackbar(...)	Add trackbar (slider) to the specified window
setMouseCallback(...)	Set the callback on mouse clicks and movements in the specified window

See camshiftdemo.cpp and other OpenCV samples on how to use the GUI functions.

Camera Calibration, Pose Estimation and Depth Estimation

calibrateCamera()	Calibrate camera from several views of a calibration pattern.
findChessboardCorners()	Find feature points on the checkerboard calibration pattern.
solvePnP()	Find the object pose from the known projections of its feature points.
stereoCalibrate()	Calibrate stereo camera.
stereoRectify()	Compute the rectification transforms for a calibrated stereo camera.
initUndistortRectifyMap()	Compute rectification map (for remap()) for each stereo camera head.
StereoBM, StereoSGBM	The stereo correspondence engines to be run on rectified stereo pairs.
reprojectImageTo3D()	Convert disparity map to 3D point cloud.
findHomography()	Find best-fit perspective transformation between two 2D point sets.

To calibrate a camera, you can use calibration.cpp or stereo_calib.cpp samples. To get the disparity maps and the point clouds, use stereo_match.cpp sample.

Object Detection

matchTemplate	Compute proximity map for given template.
CascadeClassifier	Viola's Cascade of Boosted classifiers using Haar or LBP features. Suits for detecting faces, facial features and some other objects without diverse textures. See facedetect.cpp
HOGDescriptor	N. Dalal's object detector using Histogram-of-Oriented-Gradients (HOG) features. Suits for detecting people, cars and other objects with well-defined silhouettes. See peopledetect.cpp

استفاده از این تحقیق با ذکر منبع بلامانع است .

یا حق

Ehsanavr.com