# DOCLIB – A Collaborative Approach to Document Image Processing




Booz | Allen | Hamilton

**Kevin Chen**
**Summit Sampat**

**David Doermann**
**Guangyu Zhu**

**Stefan Jaeger**

**Booz | Allen | Hamilton**
**134 National Business Pkwy**
**Annapolis Junction, MD 20701**
**chen_kevin@bah.com**

**Department of Defense**

**University of Maryland,**
**College Park, MD 20742, USA**
**jaeger@umiacs.umd.edu**

# Often times, valuable intellectual capital is lost due to technology transfer challenges

- **Software development environment and dependencies (compilers, Makefile, etc)**
- **Dependencies**
  - **External software**
  - **Hardware**
  - **Platform**
- **Limited resources**
- **Diverse skill sets**

- **Example**
  - **Transferring a new software algorithm from a University research group to a Government agency.**
    - **University:**
      - **Build using MSVC**
      - **Using MFC**
      - **libjpeg version X.XX**

    - **Government agency :**
      - **Build using Linux g++**
      - **Libjpeg version X.XX.Y**

# A common model will facilitate technology transfer

- **Common Framework**
  - **Eases technology transfer**
  - ***Ensures software compatibility***
- **Scalable**
  - ***Accommodates frequently changing requirements***
  - **Eases R & D of new algorithms**
  - **Enables simple algorithm comparison**
- **Robust and Stable**
  - **Enables high quality standards**
  - ***Prepares Govt.  to transfer technology***
- **Communication**
  - **Facilities technology transfer**
  - **Fosters collaboration**
  - ***Balances the needs of academia, Government, and industry***

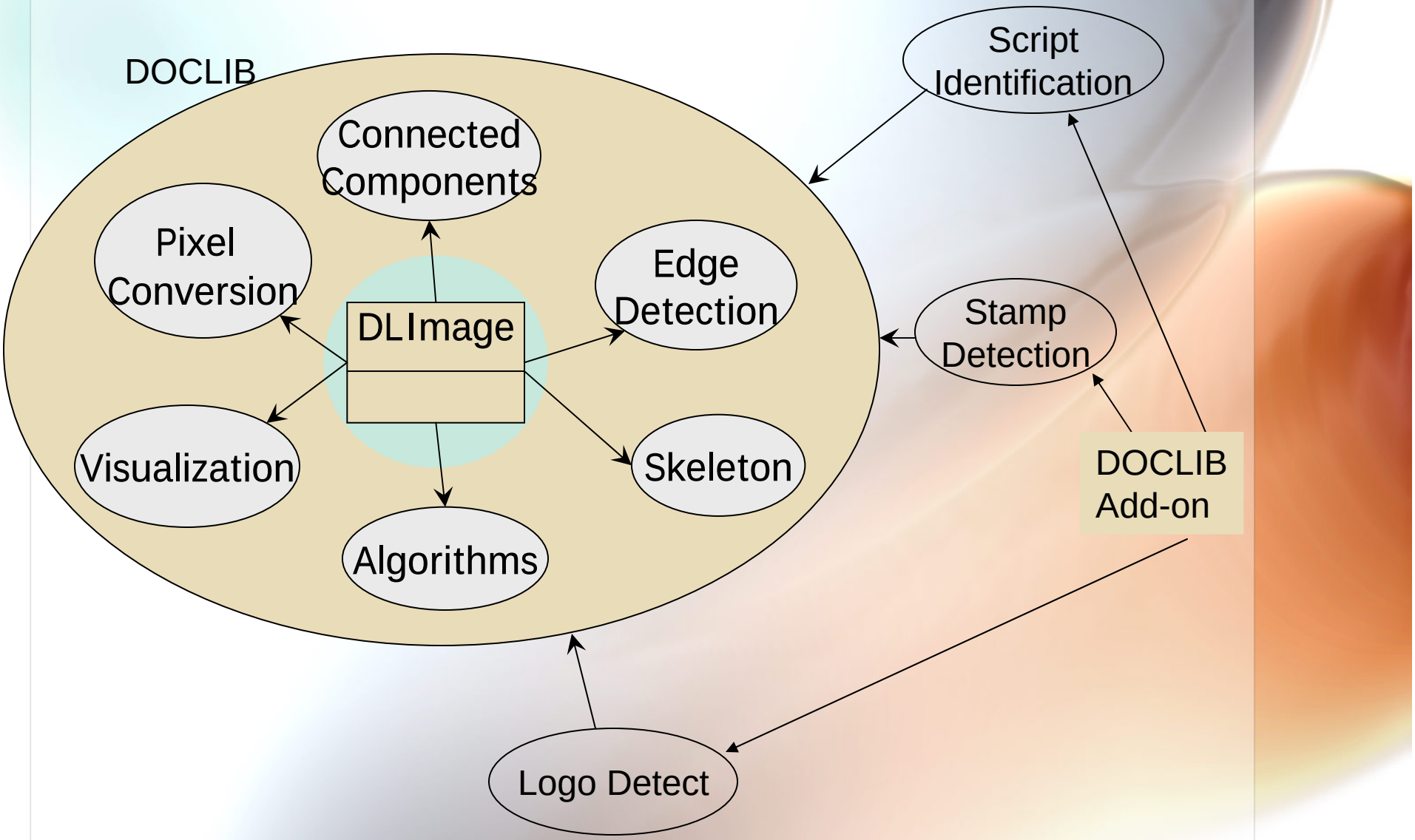
# Several tools and techniques were established to aid technology transfer and enhance performance

- **Software Development Process**
- **Shared Development Server**
- **Common Development Environment**
  - **Compiler**
  - **Supported platforms**
  - **Coding Standards**
- **Developer & User Web Portal**
  - **Distribution of software releases**
  - **Software Tutorial (example code)**
  - **Documents/Presentations/Design Archives**
- **Bug Tracker**
  - **Central Bug Repository**
  - **Task Delegation**
- **Concurrent Version System (CVS)**

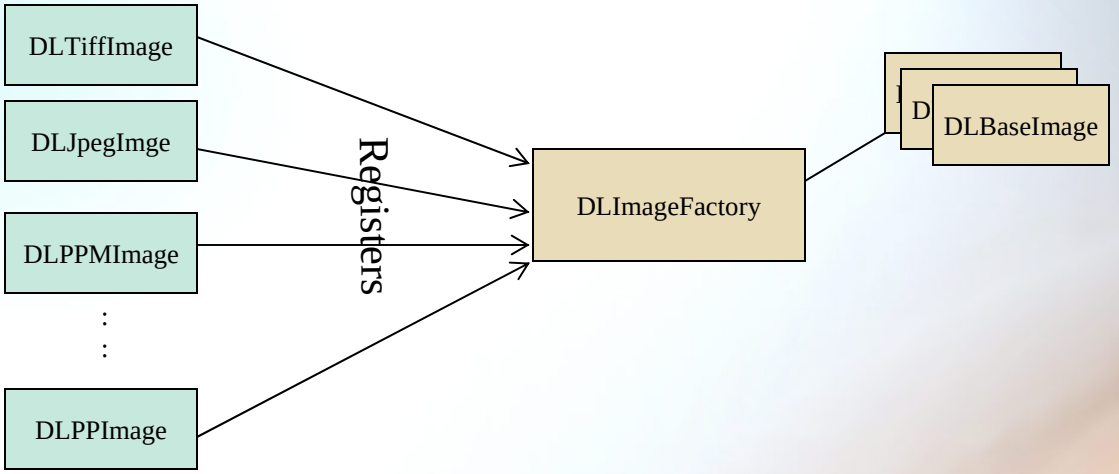
# These tools and techniques were applied throughout a collaborative software development effort

- **DOCLIB**
  - **Image & Document Processing Library**
  - ***Tailors to Government needs***
  - **Joint, Collaborative Effort**
    - **Government**
    - **Booz | Allen | Hamilton**
    - **University of Maryland, College Park**
  - **Research Facilitation Vehicle**
    - **Used successfully in many Government and Academic applications**
    - **DOCLIB, a Core Library with an Add-on Construct**
  - **Easy to use**
  - **Designed to appeal to non-object oriented developers**

# The DOCLIB system architecture is easily extendable



# DOCLIB image types plug and play into the existing architecture

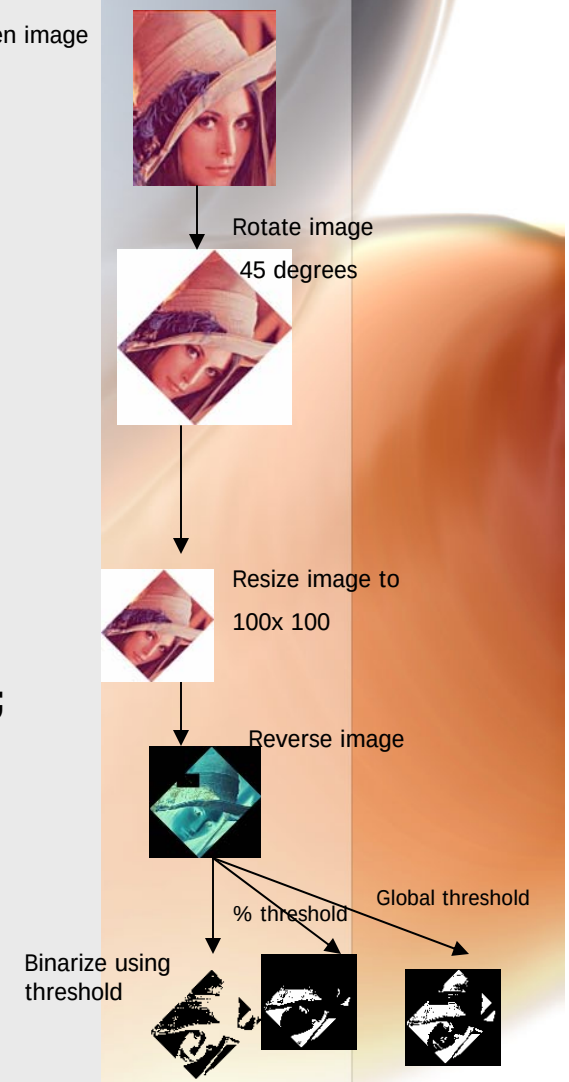



Design Factors:

- Image Type objects are static/singleton objects created on startup
- DLImageFactory is a static/singleton object
- Image Type objects registers itself with the DLImageFactory during startup
- DLImageFactory keeps a list of supported Image objects as each image type calls the register function
- Additional image types can be plugged into DOCLIB without modifying existing DOCLIB code.

# DOCLIB makes it easy to code and swap algorithms to test performance

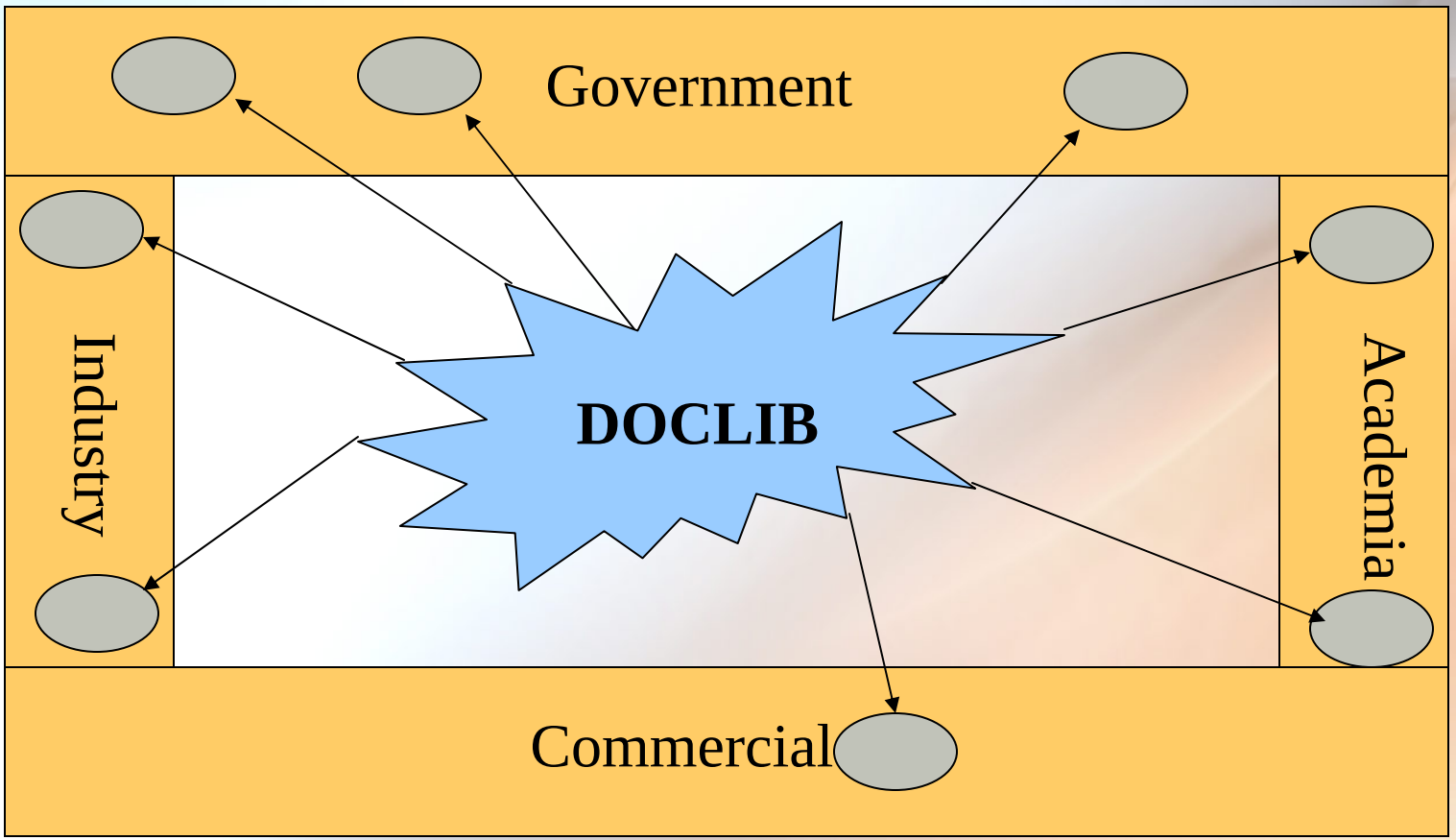
```
int main(int argc, char *argv[]) {
        // open tif image
        DLImage image("..\\..\\TestImage\\lena.tif");

        //rotate image (a) 45 dregrees
        image = image.dlRotateImage(45);

        //resize image to 100x 100
        image = image.dlResizeImage(100,100);

        // reverse image
        image = image.dlReverseImage();

        //convert to binary image using a threshold
        DLImage image_bin_thres =
            dlDownscaleColor2Binary_threshold(image, 100. 100, 100);

        //Want to perform percent threshold binarization?
        DLImage image_bin_percent =
             dlDownscaleColor2Binary_percentThreshold(image, 0.60);

        //binarization routine
        DLImage image_bin_global =
              dlDownscaleColor2Gray_global(image);

} //end main
```



# DOCLIB currently supports several image processing features

1. Open TIFF, JPEG, GIF, PPM, PNG, BMP, PBM(P1 and P4)
    1a. Memory based read TIFF
    1b. Memory based read GIF
    1c. Memory based read PNG
    1d. Memory based read BMP
    1e. Memory based read PPM
    1f. Memory based read JPEG
    1g. Memory based read PBM
    1h. Supports multiple page TIFF file
2. Save TIFF, JPEG, GIF, PPM, and PNG images to disk
    2a. Memory based write TIFF
    2b. Memory based write BMP
3. Calculate connected components
4. Save individual components of an image to disk
5. Resize an image
6. Rotate an image (large image takes long time)
7. Copy an image
8. Extract sub-image
9. Flip image
10. Contour image
11. Reverse image
12. Paste Image
13. Set Pixel of an image
14. Convert images to and from BIT(1 bit, black/white), BYTE(8 bits, gray), COLOR (24 bits)
15. Draw shapes within images
    a. Line (color)
    b. box (color)
16. Dilate image
17. Erode image
18. Sharpen image
19. Blur image
20. Mask image
21. Convert to YCrCb
22. YCrCbBinarization
23. 256Color quantized
24. PercentThresholdBinarization percent (i.e 90 or .9)
25. ThresholdBinarization rThresh gTresh bThresh
26. Color2Gray_global
27. Gray2Binary_global thresh
28. Binary2Color
29. Gray2Color
30. Binary2Gray
31. loadDocFile - Load an image as a document. Flips bits if black pixels are more than white pixels.
32. Histgram for 1 and 8 bit images
33. Projection
34. Skeletonize
35. deskew
36. Loading an unknown image from memory or from file (Unknown image type must be one of the supported images)
37. Skeletonize
38. DLDocument
39. Centroid Calculation

# DOCLIB continues to be developed and distributed across academia, industry, and Government domains



# Application of collaboration tools and techniques facilitates technology transfer

- **Allowing researchers to focus on their core competencies**
- **Saving time and frustration**
- **Increasing collaboration efforts**
- **Aligning needs across domains**
- **Minimizing debugging time**
- **Etc…**

# Questions?

## Point of Contacts

**David Doermann**
**doermann@umiacs.umd.edu**
**University of Maryland, College Park**

**Kevin Chen**
**chen_kevin@bah.com**
**Booz | Allen | Hamilton**

**Stefan Jaeger**
**jaeger@umiacs.umd.edu**
**University of Maryland, College Park**